Kerberos Server Version 3.1 Administrator's Guide

HP-UX 11i v2

Edition 5

Manufacturing Part Number: T1417-90009
E0905

United States

© Copyright 2005 Hewlett-Packard Development Company, L.P.

Legal Notices

The information contained herein is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

U.S. Government License

Proprietary computer software. Valid license from HP required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Trademark Notices

UNIX is a registered trademark in the United States and other countries, licensed exclusively through The Open Group.

MS-DOS, Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation in the United States and other countries.

Copyright Notices

© Copyright 2001-2005 Hewlett-Packard Development Company L. P.

© Copyright 1979, 1980, 1983, 1985-93 Regents of the University of California

This software is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California.

© Copyright 1983-2005 Hewlett-Packard Co., All Rights Reserved
© Copyright 1979, 1980,1983, 1985-1993 The Regents of the Univ. of California
© Copyright 1980, 1984, 1986 Novell, Inc.
© Copyright 1986-1992 Sun Microsystems, Inc.
© Copyright 1985-2002 Massachusetts Institute of Technology
© Copyright 1989-93 The Open Software Foundation, Inc.
© Copyright 1986 Digital Equipment Corporation
© Copyright 1990 Motorola, Inc.
© Copyright 1990, 1991, 1992 Cornell University
© Copyright 1989-1991 The University of Maryland
© Copyright 1988 Carnegie Mellon University
© Copyright 1984-2002 FairCom Corporation
© Copyright 1998-2002 Cybersafe Corporation
© Copyright 1991-2002 Mentat, Inc.
© Copyright 1996 Morning Star Technologies, Inc.
© Copyright 1996 Progressive Systems, Inc.
© Copyright 1991-2000 Isogon Corporation, All Rights Reserved.
© Copyright 1996 OpenVision Technologies, Inc., All Rights Reserved.

Contents

1. Overview

Introduction 25
How the Kerberos Server Works 26
Authentication Process 27
DES Versus 3DES Key Type Settings. 31
Introduction to LDAP 32
LDAP Advantages. 32
Integrating Kerberos Server v3.1 with LDAP. 33
How is the Kerberos Principal Integrated in to the LDAP Directory?. 34

2. Installing the Kerberos Server v3.1

Prerequisites 37
System Requirements 38
Hardware Requirements 38
Software Requirements 38
Version Compatibility 38
Installing the Server 39

3. Migrating to a Newer Version of the Kerberos Server

Migrating from Kerberos Server Version 1.0 to 3.0 43
Migrating from Kerberos Server Version 2.0 to Version 3.0 47
Migrating from Kerberos Server Version 3.0 to Version 3.1 49

4. Interoperability with Windows 2000

Understanding the Terminology 53
Kerberos Server and Windows 2000 Interoperability 55
Establishing Trust Between Kerberos Server and Windows 2000 56
Single Realm (Domain) Authentication 58
Interrealm (Interdomain) Authentication 59
Special Considerations for Interoperability 60
Database Considerations 60
Encryption Considerations. 60
Postdated Tickets 60

5. Configuring the Kerberos Server With C-Tree Backend

Contents

Configuration Files for the Kerberos Server 64
The krb.conf File 65
The krb.conf File Format 65
The krb.realms File. 66
The krb.realms File Format 67
Autoconfiguring the Kerberos Server 69
Configuring the Kerberos Server with C-Tree 71

6. Configuring the Kerberos Server with LDAP

Configuration Files for LDAP Integration 74
The krb5_ldap.conf File 74
The krb5_ldap.conf File Format 75
The krb5_schema.conf File. 77
The krb5_schema.conf File Format 78
The krb5_map.conf File 81
The krb5_map.conf File Format 81
Planning Your LDAP Configuration 83
Before You Begin. 83
Setting up Your LDAP Configuration 84
Autoconfiguring the Kerberos Server With LDAP Integration 88
Configuring the Kerberos Server with LDAP 88
Manually Configuring the Kerberos Server with LDAP 92
Editing the Configuration Files 92

7. Configuring the Primary and Secondary Security Server

Configuring the Primary Security Server. 96
Create the Principal Database After Installation. 96
Add an Administrative Principal 97
To add an Administrative Principal Using the HP Kerberos Administrator 97
To Add an Administrative Principal Through the Command Line 98
Create the host/<fqdn> Principal and Extracting the Service Key 98
Start the Kerberos Daemons 99
Define Secondary Security Server Network Locations 100
Security Policies 101
Password Policy File 101
The admin_acl_file 101

Starting the Security Server 102
Configuring the Secondary Security Servers with C-Tree. 103
Creating the Principal Database 103
Copying the Kerberos Configuration File 103
Creating a host/<fqdn> Principal and Extracting the Key. 104
Configuring the Secondary Security Servers with LDAP 105
Copying the Kerberos Configuration File 105
Creating a stash file using the kdb_stash utility 105
Using Indexes to Improve Database Performance 107

8. Administering the Kerberos Server

Administering the Kerberos Database 111
The kadmind Command 112
The admin_acl_file File 113
Assigning Administrative Permissions 114
Adding Entries to admin_acl_file. 116
Creating Administrative Accounts 117
Using Restricted Administrator. 117
How the r/R Modifiers Work 117
Password Policy File. 119
Editing the Default File 119
Principals 121
Adding User Principals. 123
Adding New Service Principals 123
Reserved Service Principals 124
Removing User Principals 127
Removing Special Privilege Settings 127
Protecting a Secret Key. 128
Removing Service Principals 128
The kadmin and kadminl Utilities 130
Administration Utilities 131
HP Kerberos Administrator 132
Standard Functionality of the Administrator 133
Local Administrator – kadminl_ui 134
Using kadminl_ui 134
Principals Tab. 136

Contents

General Tab (Principal Information Window) 139
Adding Principals to the Database 143
  Adding Multiple Principals with Similar Settings 145
Creating an Administrative Principal 146
Searching for a Principal 149
Deleting a Principal 151
Loading Default Values for a Principal. 152
Restoring Previously Saved Values for a Principal 153
Changing Ticket Information 154
Rules for Setting Maximum Ticket Lifetime 155
Rules for Setting Maximum Renew Time. 156
Changing Password Information. 158
Password Tab (Principal Information Window) 160
  Change Password Window (Password Tab). 162
Changing a Key Type. 165
  Changing a DES-CRC or DES-MD5 Principal Key Type to 3DES 165
Changing Principal Attributes 167
Attributes Tab (Principal Information Window) 168
LDAP Attributes Tab (Prinicpal Information Window). 175
Deleting a Service Principal 177
Extracting Service Keys. 178
Extracting a Service Key Table 180
Using Groups to Control Settings 182
  Editing the Default Group 182
Group Information Window (Principal Information Window). 184
  Principal Attributes 186
    Setting the Default Group Principal Attributes 186
    Default Principal Attributes. 186
Setting Administrative Permissions 188
Administrative Permissions 189
Realms Tab 193
  Realm Information Window 195
Adding a Realm 196
Deleting a Realm 197
Remote Administrator – kadmin_ui 198
Manual Administration Using kadmin. 202

Adding a New Principal 204
Adding a Random Key 205
Specifying a New Password 205
Changing Password to a New Randomly Generated Password 206
Deleting a Principal 206
Extracting a Principal 207
Listing the Attributes of a Principal 208
Modifying a Principal 208
Number of Authentication Failures (fcnt) 209
Key Version Number Attribute 210
LDAP DN 210
Policy Name 211
Attributes 211
Allow Postdated Attribute 211
Allow Renewable Attribute 212
Allow Forwardable Attribute 213
Allow Proxy Attribute 214
Allow Duplicate Session Key Attribute 215
Require Preauthentication Attribute 216
Require Password Change Attribute 216
Lock Principal Attribute 217
Allow As Service Attribute 218
Require Initial Authentication Attribute 219
Set As Password Change Service Attribute 220
Password Expiration Attribute 221
Principal Expiration Attribute 222
Key Type Attribute 223
Principal Database Utilities 224
Kerberos Database Utilities 225
Database Encryption 227
Database Master Password 228
Destroying the Kerberos Database 229
Dumping the Kerberos Database 231
Loading the Kerberos Database 232
Stashing the Master Key 233
Starting and Stopping Daemons 235

Contents

Maintenance Tasks 236
Protecting Security Server Secrets 236
host/fqdn@REALM 236
Master Password 236
Backing Up primary security server Data 237
Backing Up the Principal Database 237
Removing Unused Space from the Database 239

9. Propagating the Kerberos Server

Propagation Hierarchy 243
Propagation Relationships 243
Service Key Table 244
Maintaining Secret Keys in the Key Table File 244
Extracting a Key to the Service Key Table File 244
Creating a New Service Key Table File 245
Deleting Older Keys from the Service Key Table File 245
Propagation Tools 246
The kpropd Daemon 248
The mkpropcf Tool 249
The kpropd.ini File 251
Sections 252
The [default_values] Section 252
The [secsrv_name] Section 254
Examples 255
The prpadmin Administrative Application 257
Setting Up Propagation 258
Monitoring Propagation 263
Monitoring the Log File 263
Critical Error Messages 263
Monitoring Propagation Queue Files 264
Monitoring Old File Date and Large File Size 264
Updating the principal.ok Time Stamp 265
Comparing the Database to Its Copies 265
The kdb_dump Utility 267
Restarting Propagation Using a Simple Process 268
Restarting Propagation Using the Full Dump Method 268

Propagation Failure 269
Converting a secondary security server to a primary security server 270
Restarting Services 271
Cleaning the Temp Directory 271
Configuring Multirealm Enterprises 272
Number of Realms per Database 272
primary security servers Supporting Multiple Realms 272
Multiple primary security servers Supporting a Single Realm 273
Adding More Realms to a Multirealm Database. 273
Database Propagation for Multirealm Databases. 274

10. Managing Multiple Realms

Considering a Trust Relationship 277
One-Way Trust 277
Two-Way Trust 277
Hierarchical Trust 278
Other Types of Trust. 278
Configuring Direct Trust Relationships 279
Hierarchical Interrealm Trust 281
Hierarchical Chain of Trust 281
Hierarchical Interrealm Configuration 282
Configuring the Local Realm 284
Configuring the Intermediate Realm 285
Configuring the Target Realm 286

11. Troubleshooting

Characterizing a Problem 291
Diagnostic Tools Summary 293
Troubleshooting Kerberos 294
Error Messages 294
Logging Capabilities 294
UNIX Syslog File 295
Services Checklist 296
Troubleshooting Techniques. 296
General Errors 303
Forgotten Passwords 303

Contents

Locking and Unlocking Accounts 304
Clock Synchronization 304
User Error Messages 305
Decrypt Integrity Check Failed 305
Password Has Already Been Used or Is Too Close to Current One 305
Administrative Error Messages 306
Password Has Expired While Getting Initial Ticket 306
Service Key Not Available While Getting Initial Ticket. 306
Reporting Problems to Your HP Support Contact 308
The services File. 317

Tables

Table 1. HP-UX 11i Releases 20
Table 2. Publishing History Details 20
Table 4-1. Table of Analogous Terms. 54
Table 5-1. Security Server Files That Require Configuration 64
Table 5-2. Wildcard Characters 68
Table 6-1. LDAP Configuration Files 74
Table 6-2. krb5_ldap.conf File Format 75
Table 8-1. Configuration Files Required for kadmind 112
Table 8-2. Administrative Permission Settings 114
Table 8-3. Default Password Policy Settings for the Base Group 119
Table 8-4. Administration Utilities 131
Table 8-5. Function of OK, Apply, and Cancel Buttons 133
Table 8-6. Principals Tab Components 137
Table 8-7. Principal Information Window Components 139
Table 8-8. General Tab Components 140
Table 8-9. Search Criteria 149
Table 8-10. Password Tab Components. 160
Table 8-11. Change Password Window Components 163
Table 8-12. Attributes Tab Components 168
Table 8-13. Extract Service Key Table Components 181
Table 8-14. Group Information Window Components 185
Table 8-15. Group Information Window Components 190
Table 8-16. Realms Tab Components 194
Table 8-17. Realm Information Window Components 195
Table 8-18. Require Initial Authentication Attribute Settings 220
Table 8-19. Principal Database Utilities. 224
Table 8-20. Starting and Stopping Daemons and Services 235
Table 9-1. Propagation Tools 246
Table 9-2. primary security server Services and Daemons 259
Table 11-1. Diagnostic Tools 293
Table 11-2. Troubleshooting Scenarios 296
Table 11-3. Troubleshooting Scenarios for your LDAP-based Kerberos server . . . 299
Table A-1. Configuration Worksheet 312

Tables

Table A-2. Configuration Worksheet Explanation 312

Figures

Authentication Process 28
Integrating a Kerberos Principal in to the LDAP Directory 34
Principals Tab 137
Principal Information Window 139
Change Password Window 144
Administrative Permissions Window 147
Password Tab 160
Change Password Window 163
Attributes Tab 168
LDAP Attributes Tab 176
Extract Service Key Table Window 180
Group Information Window 185
Administrative Permissions Window 189
Realms Tab 194
Realm Information Window 195
Logon Screen 199
Change Password Screen 200
Warning Message 200
Hierarchical Interrealm Configuration 283

Figures

About This Manual

This manual describes how to install, configure, administer, and troubleshoot the Kerberos server on HP Integrity servers running the HP-UX 11i v2 operating system.

Intended Audience

HP intends this manual for system managers or administrators responsible for configuring and maintaining the Kerberos server running HP-UX 11i v2.

This manual is based on the assumption that you meet the following prerequisites:

- Understand distributed network concepts and client/server computing
- UNIX® operating system
- Understand the Kerberos basics
- Understand LDAP concepts

What Is in This Document

Kerberos Server Version 3.1 Administrator's Guide is divided into the following chapters, which contain information about installing, configuring, and administering the Kerberos server:

- *Chapter 1, "Overview," on page 23*: Provides an introduction to the Kerberos server, outlines the new features in this release, and highlights the key advantages of using the Kerberos server. It also provides an introduction to LDAP, and provides details on integrating Kerberos v3.1 with LDAP.
- *Chapter 2, "Installing the Kerberos Server v3.1," on page 35*: Describes how to install the Kerberos server on the HP-UX 11i v2 operating system.
- *Chapter 3, "Migrating to a Newer Version of the Kerberos Server," on page 41*: Explains the migration process from earlier versions of the Kerberos server to v3.1.

- *Chapter 4, “Interoperability with Windows 2000,” on page 51*: Contains information specific to establishing interoperability with Windows 2000® Kerberos implementations.
- *Chapter 5, “Configuring the Kerberos Server With C-Tree Backend,” on page 63*: Provides information on the configuration files required to configure the Kerberos server with C-tree as the backend database.
- Chapter 6, “Configuring the Kerberos Server with LDAP,” on page 73: Provides information on the configuration files required to configure the Kerberos server with LDAP as the backend database.
- Chapter 7, “Configuring the Primary and Secondary Security Server,” on page 95: Describes the procedure for configuring the primary and secondary servers.
- *Chapter 8, “Administering the Kerberos Server,” on page 109*: Describes the procedures for administering the Kerberos server database. It also discusses principals and their attributes.
- *Chapter 9, “Propagating the Kerberos Server,” on page 241*: Describes how to propagate the Kerberos server database from the primary security server to the secondary security servers.
- *Chapter 10, “Managing Multiple Realms,” on page 275*: Explains interrealm authentication and interoperability trust. In addition, it gives you an overview of the additional server configuration requirements in deployments that use multiple realms and interrealm authentication.
- *Chapter 11, “Troubleshooting,” on page 289*: Describes how to troubleshoot the common problems encountered while using the Kerberos server. In addition, it contains a brief note on reporting problems to your Hewlett-Packard Support Contact.
- Appendix A, “Configuration Worksheet,” on page 311: Provides a worksheet that will help you configure the Kerberos server with LDAP as the backend database.
- Appendix B, “Sample krb.conf File,” on page 317: Provides a sample `krb.conf` file.
- Appendix C, “Sample krb.realms File,” on page 319: Provides a sample `krb.realms` file.
- **Glossary**

- **Index**

Typographic Conventions

The following conventions are used throughout this manual:

Text Conventions

italic	Identifies book titles.
bold	Identifies options, command buttons, and menu items.

Syntax Conventions

`fixed width`	Identifies file names, system prompts, operating system commands, and UNIX error and system messages.
italic fixed width	Identifies variables that you need to replace according to your environment.
bold fixed width	Identifies the default in a series of parameters.
|	Separates mutually exclusive parameters; only one of the parameters separated by the bar is allowed.
[ ]	Indicate that the enclosed parameters are optional.
{ }	Indicate that only one of the enclosed parameters are required.
\	Indicates that a command line, parameter, or code continues on the following line.
#	Precedes a UNIX command that must be performed as a root user.
%	Precedes a UNIX command that must be performed as an ordinary user.

HP-UX Release Name and Release Identifier

Each HP-UX 11i release has an associated release name and release identifier. The `uname`(1) command with the `-r` option returns the release identifier. Table 1 lists the releases available for HP-UX 11i.

Table 1 HP-UX 11i Releases

Release Identifier	Release Name	Supported Processor Architecture
B.11.11	HP-UX 11i v1	PA-RISC
B.11.20	HP-UX 11i v1.5	Intel® Itanium®
B.11.22	HP-UX 11i v1.6	Intel Itanium
B.11.23	HP-UX 11i v2	Intel Itanium

Publishing History

Table 2 provides, for a particular document, the manufacturing part number, the respective operating systems, and the publication date.

Table 2 Publishing History Details

Document Manufacturing Part Number	Operating System Supported	Publication Date
T1417-90001	HP-UX 11.0 and 11i v1	September 2001
T1417-90003	HP-UX 11.0 and 11i v1	June 2002
T1417-90007	HP-UX 11i v2	October 2003
T1417-90009	HP-UX 11i v2	April 2004

Related Software Products

Following are the products related to the Kerberos server:

- PAM Kerberos on HP-UX 11i v2, delivered as part of the HP-UX Internet operating environment component.

- KRB5 Client Software on HP-UX 11i v2, delivered as part of the core operating system.
- GSS-API on HP-UX 11i v2, delivered as part of the core operating system.

Related Documentation

For more information on the Kerberos server, see the following manuals:

- *Configuration Guide for Kerberos Client Products on HP-UX* (T1417-90006)
- *PAM Kerberos Release Notes for HP-UX 11i v2* (J5849-90011)
- *PAM Kerberos Release Notes for HP-UX 11i* (J5849-90002)
- *KRB5 Client Software Release Notes for HP-UX 11.0* (J5849-90005)
- *GSS-API Release Notes for HP-UX 11.0* (J5849-90006)
- *HP-UX Internet Services Administrator's Guide* (B2355-90774)
- *Using HP-UX Internet Services* (B2355-90827)
- *HP-UX 11i v2 Installation and Update Guide* (for HP-UX operating system v11i v1 or later) (5187-2725)

Accessing the World Wide Web

See the following web sites for more information on the Kerberos server:

- HP Technical Documentation and White Papers
  - `http://docs.hp.com`
  - `http://www.unixsolutions.hp.com/products/hpux/hpux11/whitepapers/netsecur.pdf`
- HP-UX IT Resource Center
  - `http://us-support.external.hp.com` (America and Asia Pacific)
  - `http://europe-support.external.hp.com` (Europe)

Related Request for Comments (RFCs)

See the following RFCs for more information on the Kerberos server:

- RFC 1510 - *The Kerberos Network Authentication Service (V5)*
- RFC 1964 - *The Kerberos v5 GSS-API Mechanism*
- RFC 2743 - *Generic Security Service Application Program Interface*
- RFC 2744 - *Generic Security Service API*

You can access these RFCs at the following Web site:

`http://www.ietf.org/rfc.html`

HP Encourages Your Comments

HP welcomes any comments and suggestions you have on this manual.

You can send your comments in the following ways:

- Internet electronic mail: `netinfo_feedback@cup.hp.com`
- Using a feedback form located at the following URL:

  `http://docs.hp.com/assistance/feedback.html`

Please include the following information along with your comments:

- The full title of the manual and the part number. (The part number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of HP-UX that you are using.

1 Overview

This chapter provides an introduction to the Kerberos server v3.1, available on the HP-UX 11i v2 operating system.

This chapter discusses the following topics:

- "How the Kerberos Server Works" on page 26
- "Authentication Process" on page 27
- "DES Versus 3DES Key Type Settings" on page 31
- "Introduction to LDAP" on page 32
  - — "Integrating Kerberos Server v3.1 with LDAP" on page 33

Introduction

The term **Kerberos** was derived from the Greek mythology. Cerberus is the latin variant of Kerberos, who guarded the entrance of Hades, the Greek hell. The Kerberos security system, on the other hand, guards electronic transmissions that are sent across a network.

Kerberos is a mature network authentication protocol based on the RFC 1510 (*The Kerberos Network Authentication Service (V5)*) specification of the Internet Engineering Task Force (IETF). It is designed to provide strong authentication for client or server applications using the shared secret key cryptography.

The Kerberos server is based on a distributed client/server architecture. It ensures secure communication in a networked environment by leveraging individual trust relationships. It then brokers that trust across enterprise wide, distributed client/server networks.

How the Kerberos Server Works

The basic currency of Kerberos is the ticket, which the user presents to use a specific service. Each service, be it a login service or an FTP service, requires a different kind of ticket. The applications on the Kerberos server keep track of all the various kinds of tickets.

When you first log on to Kerberos each day, you enter your Kerberos password. In return, the Kerberos server gives you an initial ticket, which you use to request additional tickets from the Kerberos server for all the other services. For this reason, the initial ticket is also called the ticket-granting ticket, or TGT.

Use the Kerberos protocol to secure the communication between the client and server. Thus, client programs make authentication requests to an authentication server, and server programs in turn service those client requests. Based on your user credentials, the server program grants or denies your request to access network applications and services. The Kerberos server allows entities to authenticate themselves, without having to transmit their passwords in clear text form over the network.

For more information on the basics of Kerberos, see *Installing, Configuring and Administering the Kerberos Server on HP-UX 11i* (T1417-0001), available at `http://www.docs.hp.com/hpux/internet/index.html#Kerberos`.

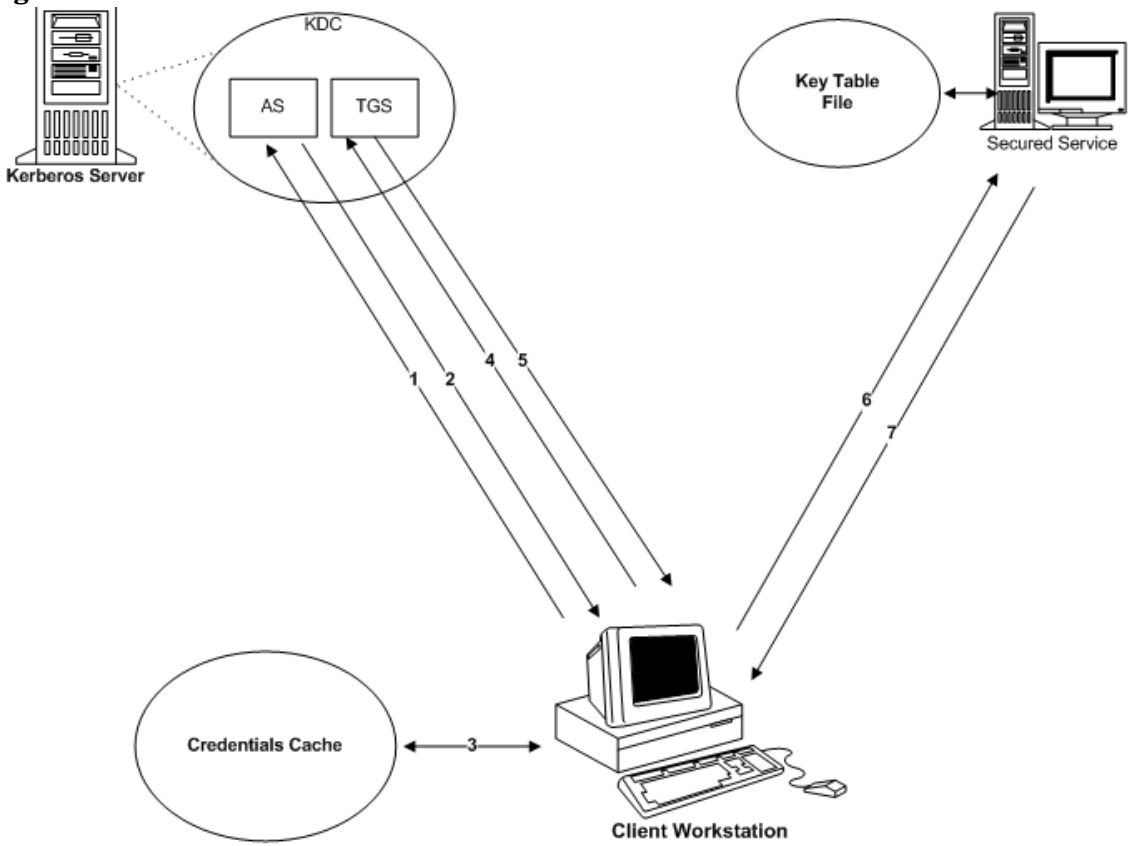
Authentication Process

The Kerberos server grants tickets to your user principal to access secured network services. You must log on to the server by providing your user name and password. When the server authenticates you, it returns a set of initial credentials for you, including a TGT and a session key.

The Kerberos server grants a service ticket for a specific service principal that can be associated with one or more Kerberos-secured services on the same system. A client application uses your service ticket to authenticate you to a Kerberos-secured network service. The secured client application automatically handles the transactions with the server and the secured application server. Service tickets and associated session keys are generally cached in your user credentials cache along with the TGT of the user.

Figure 1-1 illustrates the actions of the components and the Kerberos protocol in a secured environment.

Figure 1-1 Authentication Process



The following is a description of how a client and server authenticate each other using Kerberos:

Step 1. You can begin to use a Kerberos-secured application by entering your principal name and password. Optionally, you can request specific ticket flags and specify the key type to be used to construct the secret key. You can also accept the default values configured for the client.

You can send the following information to the Authentication Service (AS) to obtain credentials:

- Client-indicates the user name, also referred to as the principal name
- Server-indicates the Application Server
- Time stamp
- Nonce

Step 2. If the AS decrypts the message successfully, it authenticates the requesting user and issues a TGT. The TGT contains the user name, a session key for your use, and name of the server to be used for any subsequent communication. The reply message is encrypted using your secret key.

Step 3. The client decrypts the message using your secret key. The TGT and the session key from the message are stored in the client's credential cache. These credentials are used to obtain tickets for each network service the principal wants to access.

The Kerberos protocol exchange has the following important features:

- The authentication scheme does not require that the password be sent across the network, either in encrypted form or in clear text.
- The client (or any other user) cannot view or modify the contents of the TGT.

Step 4. To obtain access to a secured network service such as `rlogin`, `rsh`, `rcp`, `ftp`, or `telnet`, the requesting client application uses the previously obtained TGT in a dialogue with the TGS to obtain a service ticket. The protocol is the same as used while obtaining the TGT, except that the messages contain the name of the server and a copy of the previously obtained TGT.

Step 5. The TGS returns a new service ticket that the application client can use to authenticate the service.

Step 6. The application client tries to authenticate to the service on the application server using the service ticket obtained from the TGS.

The secure application validates the service ticket using the service key of the server that is present in the key tab file. Using the session key, the server decrypts the authenticator and verifies the identity of the user. It

also verifies that the user's service ticket has not expired. If the user does not have a valid service ticket, then the server will return an appropriate error code to the client.

Step 7. (Optional) At the client's request, the application server can also return the timestamp sent by the client, encrypted in the session key. This ensures a mutual authentication between the client and the server.

DES Versus 3DES Key Type Settings

In the processes outlined in the section “Authentication Process” on page 27, if the user principal and the service principal do not use the same key type, the process continues as described.

The Kerberos server acts as the only trusted party, and the client or the service does not accept a message encrypted by the client or the service key. Both the client application and the service share a secret key only with the server.

The authenticator data that the service and client encrypt or decrypt is encrypted in session keys. The server sends the required session keys to the client and service in packets that are encrypted with their respective keys. The Kerberos server checks the key type settings for the user principals and service principals and determines the most secure encryption allowed for the session key. If the user principals and service principals have a 3DES key stored in the database, the session key type that is returned is of type 3DES. If only one has a 3DES key and the other has a DES key, then the session key that is returned is of type DES.

The server never returns a session key in the service ticket packet that uses stronger encryption than the session key included with a TGT packet. If a user principal has both 3DES and DES keys and uses the DES key to obtain a TGT, all service tickets obtained using this TGT contain DES session keys.

IMPORTANT

The `krbtgt/<REALM NAME>` is the ticket-granting principal. This is a reserved principal that is automatically created when you add a realm to the database. You must assign a key type for the `krbtgt/<REALM NAME>` principal or the default key, issued by the Kerberos server, uses the 3DES encryption type.

Introduction to LDAP

The Lightweight Directory Access Protocol (LDAP) is a lightweight protocol for accessing directory services. LDAP defines a message protocol used by directory clients and directory servers. It is a fast-growing technology for accessing common directory information. LDAP has been embraced and implemented in most network-oriented middleware. LDAP has gained wide acceptance as the directory access method of the Internet and is therefore becoming strategic within corporate intranets.

As the number of different networks and applications has grown, the number of specialized directories of information has also grown, resulting in islands of information that are difficult to maintain. LDAP, an open industry standard, has evolved to meet these needs of providing access to a common directory infrastructure. LDAP defines a standard method for accessing and updating information in a directory.

LDAP Advantages

LDAP has evolved as a lightweight protocol for accessing information in X.500 directory services. It has since become more independent of X.500, and servers that specifically support the LDAP protocol rather than the X.500 Directory Access Protocol. The success of LDAP has been largely due to the following characteristics that make it simpler to implement and use, compared to X.500 and DAP:

- Omits duplicate, rarely used, and esoteric features. This makes LDAP easier to understand and to implement.
- Runs over TCP/IP rather than the OSI protocol stack. TCP/IP is less resource-intensive and is widely available.
- Encodes data for transport over networks by using a simplified version of the same encoding rules that is used by X.500.
- Uses strings to represent data rather than complicated structured syntax such as ASN.1 (Abstract Syntax Notation One).

Integrating Kerberos Server v3.1 with LDAP

You can configure Kerberos server v3.1 with LDAP as the backend database. By integrating the Kerberos principals with the corresponding users in the LDAP directory, you store data for mechanisms, such as UNIX and Kerberos in a common repository. Also, you can secure user credentials by mandating users to use LDAP credentials.

Implementing this solution involves the following steps:

— Modifying the configuration files on the Kerberos server

— Extending the LDAP directory schema

The Kerberos Server v3.1 Administrator's Guide first details the design specifications in terms of the Kerberos Server requirements and the LDAP directory requirements. It then covers the actual implementation guidelines and procedures used to accomplish this solution.

You must use the `krb_2_ldap` utility to migrate your existing Kerberos database to LDAP. See "Migrating to a Newer Version of the Kerberos Server", on page 41.

You can configure your Kerberos server with LDAP by either using the autoconfiguration tool, `krbsetup`, or manually editing the LDAP configuration files located in the `/opt/krb5/examples` directory. For more information see Chapter 6, "Configuring the Kerberos Server with LDAP," on page 73. HP recommends that you use the `krbsetup` tool to configure your Kerberos server with the LDAP.

You can administer and maintain the Kerberos database by either using the HP Kerberos Administrator, a graphical user interface, or the command-line administrator. See "Administering the Kerberos Server", on page 109.

NOTE Kerberos server v3.1 supports only Netscape Directory server 6.0 (J4258CA) and later, as the LDAP backend database. You must have the LDAP-UX product installed on the Kerberos server to setup a Kerberos server with LDAP as the backend database.

How is the Kerberos Principal Integrated in to the LDAP Directory?

A directory contains a collection of objects organized in a tree structure. You can arrange entries within the DIT based on their Distinguished Names (DNs). A DN is composed of a sequence of RDNs separated by commas, such as `cn=alex,ou=R&D,o=bambi`.

Figure 1-2, displays how a Kerberos principal is integrated in to the LDAP directory.

Figure 1-2 Integrating a Kerberos Principal in to the LDAP Directory

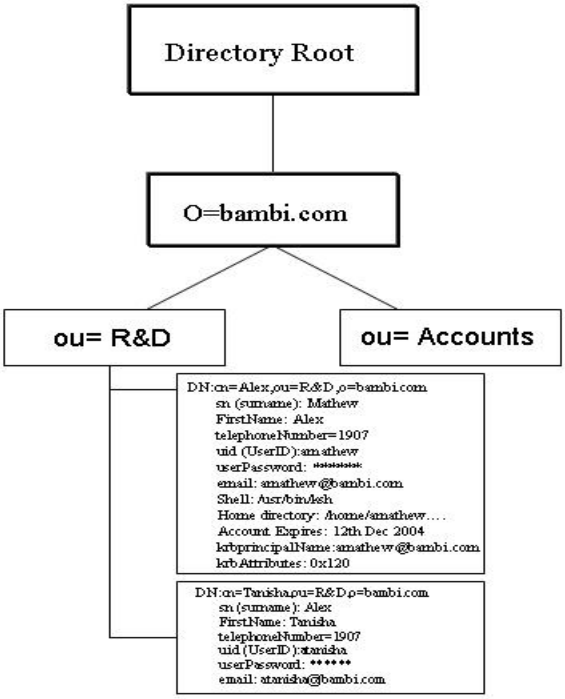


Figure 1-2 illustrates data related to the user Alex Mathew, who is located in the LDAP directory at cn=Alex, ou=accounts, o=BAMBI.COM. With both the POSIX account and LDAP information integrated, like Alex's UNIX identity, his Kerberos identity, and any other attributes related to Alex under a single LDAP directory entry.

2 Installing the Kerberos Server v3.1

This chapter describes how to install the Kerberos server v3.1 on the HP-UX 11i v2 operating system.

This chapter contains the following sections:

- “Prerequisites” on page 37
- “System Requirements” on page 38
- “Installing the Server” on page 39

Prerequisites

Before you install the server, ensure that:

- You have installed the HP-UX 11i v2 operating system on your system. To check the version of the HP-UX operating system, run the `uname -r` command at the HP-UX prompt.
- The Kerberos server is installed on a system that is physically secure and has restricted access to it. If necessary, verify that the system on which you install the Kerberos server is kept under lock and key.
- All the network services, such as `ftp`, `telnet`, `rlogin`, and `finger`, are disabled by restricting access to the machine. Edit the `/etc/inetd.conf` file to deactivate the non-kerberos services. Restart the `inetd` daemon after you modifying the `/etc/inetd.conf` file.
- The file system is protected with proper permissions in order to restrict the non root users from accessing and manipulating the files maintained by the Kerberos server, such as cache files and stash files.

System Requirements

This section describes the hardware and software requirements for the Kerberos server software for HP-UX server systems.

Hardware Requirements

The hardware requirement for installing the Kerberos server is 12 MB of free disk space.

You can install the Kerberos server v3.1 software when your system is up, and you need not reboot the system after installing the software. HP does not recommend the single-user state.

Software Requirements

Before installing the Server product, ensure that the following software is installed on your system:

- KRB5-Client Software, delivered as part of the HP-UX 11i v1 core operating system.
- LDAP-UX Client product, if you plan to enable LDAP as the backend database.
- Netscape Directory server 6.0 (J4258CA), if you plan to enable LDAP as the backend database.

For more information about any of these products, see “Related Documentation” on page 21. If you cannot find the software or information you need, contact your HP representative or log on to Website, `http://www.software.hp.com`.

Version Compatibility

The version of the Kerberos server you are installing must be compatible with the version of HP-UX you are running.

Installing the Server

To install the Kerberos server, complete the following steps:

Step 1. Insert the software media (tape or disk) in the appropriate drive.

Step 2. Type the `swinstall` command at the HP-UX prompt.

For more information on the `swinstall` command, type `man 1M swinstall` at the HP-UX prompt.

Step 3. In the Specify Source window, select the appropriate path of the depot and click **OK**.

Step 4. Highlight T1417AA in the Software Selection dialog box, and select **Mark For Install** from the Actions menu to install all filesets in the bundle.

Step 5. When you have marked the product components you want to install, select **Install (analysis)** from the Actions menu.

Step 6. When you have successfully completed the analysis, click **OK** in the Analysis dialog to load the Kerberos server filesets.

The `swinstall` utility loads the filesets. The estimated time for processing is 5 minutes.

If the installation is not successful, an error message is displayed. The cause of the failure will appear at the end of the `/var/adm/sw/swagent.log` file.

For more information, see *Managing HP-UX Software with SD-UX* at `http://www.docs.hp.com`.

Kerberos server v3.1 is ready for use.

To configure the server to act as either the primary security server or one of the secondary security servers, see Chapter 7, “Configuring the Primary and Secondary Security Server,” on page 95 for more information.

3 Migrating to a Newer Version of the Kerberos Server

This chapter describes how to migrate from the Kerberos server v1.0 to v3.0, from the Kerberos server v2.0 to v3.0, and from the Kerberos server

v3.0 to v3.1. The Kerberos database formats of v2.0 and v3.0 are compatible with each other, but the database formats of Kerberos server v1.0 and v3.0 are not compatible with each other. Therefore, migrate the database format from v1.0 to v3.0.

The Kerberos server v1.0 database contains information related both to principal and policy. However, the Kerberos server v3.0 database contains only the principal-related information, and contains the policy-related information in a separate file, `password.policy`. The Kerberos server v3.0 supports a tool to migrate the principal-related information from v1.0 to v3.0.

To migrate from the Kerberos server v2.0 database to v3.0, dump the v2.0 database using the `kdb_dump` utility, and load this dump file into the v3.0 database using the `kdb_load` utility. If you are migrating the Kerberos database from v1.0 to v3.0 or from v2.0 to v3.0, create a dump file of the older Kerberos database before installing the new version of the Kerberos server. For more information, see “Migrating from Kerberos Server Version 1.0 to 3.0” on page 43, and “Migrating from Kerberos Server Version 2.0 to Version 3.0” on page 47.

To migrate from the Kerberos server v3.0 to v3.1, dump the v3.0 database using the `krb_2_ldap` utility, and load this dump file into the v3.1 database using the `ldapmodidfy` command. For more information, see “Migrating from Kerberos Server Version 3.0 to Version 3.1” on page 49.

NOTE

You must manually migrate the policy information from v1.0 to v3.0. However, while migrating from v2.0 to v3.0, you need not migrate the `password.policy` file that contains the policy-related information.

This chapter discusses the following topics:

- “Migrating from Kerberos Server Version 1.0 to 3.0” on page 43
- “Migrating from Kerberos Server Version 2.0 to Version 3.0” on page 47
- “Migrating from Kerberos Server Version 3.0 to Version 3.1” on page 49

Migrating from Kerberos Server Version 1.0 to 3.0

If you want to use the Kerberos server with C-tree as the backend database, migrate your existing Kerberos server to Kerberos server v3.0.

In the Kerberos server v1.0, you can create a policy with any name and attribute value. Any principal can subscribe to any of the policies in the database.

In the Kerberos server v2.0, the password policy is based on the instance name of the principal. The instance name is part of the principal name. For example, in the principal, `user1/admin@hp.com`, `admin` is the instance name. The principals having the `admin` instance inherit the values defined for the `admin` policy in the `password.policy` file.

In the new version of the Kerberos server, v3.0, the password policies are based on the policy subscribed to by the principal.

The policy information is available as a dump file after you have migrated the dump file from v1.0 to v3.0. After the migration, the policy information is not migrated automatically, that is, the policy to which a principal is subscribed, is not updated in the database. The administrator needs to explicitly classify the principals and add the policies to the `password.policy` file, according to the site policy.

IMPORTANT You must modify the principals with the new policy. The instance-based rules apply if you do not specify the policy.

You need to perform the task of manually migrating the `admin_acl_file` from v1.0 to v3.0. For more information, see “The admin_acl_file File” on page 113.

To migrate from Kerberos server v1.0 to v3.0, complete the following steps:

Step 1. Dump the database on the v1.0 server.

On the Kerberos server v1.0, dump the database with the default dump version. The dump file must contain the default header, “`kdb5_util load_dump version 5`”.

```
# kdb5_util dump /opt/krb5/dumpfilev1.0
```

Step 2. Copy the dump file to the new system where you are installing the Kerberos server v3.0.

Step 3. Install the v3.0 Kerberos daemons on the new system.

Step 4. Migrate the v1.0 dump file to the v3.0 dump file.

To generate the v3.0 dump file, run the `kdb_migrate` tool on the system where Kerberos server v3.0 is installed:

```
# kdb_migrate -i /opt/krb5/dumpfilev1.0 -o
=> /opt/krb5/dumpfilev3.0 -p /opt/krb5/polv3 -1
=> /tmp/kdb_migrate.log
```

NOTE The lines beginning with => are continuations of the previous line.

If the `/var/adm/krb5/krb5kdc/kdc.conf` file does not exist and the master key name is not the default (`K/M`), specify this as an argument in `kdb_migrate` by specifying the `-M` option.

If the `/var/adm/krb5/krb5kdc/kdc.conf` file does not exist and the `-e` option is not specified, the encryption type is the encryption type of the master principal obtained from the `dumpfilev1.0`.

If the `/etc/krb5.conf` file does not exist, the migration process fails.

You can change the password of the master key while executing the migration tool. The tool prompts you for a password change. If you want to change the password, type `yes` at the command prompt. If you do not want to change the password, type `no` at the command prompt.

NOTE You must use the same password while creating the minimal database for v3.0 of the Kerberos server, as described in step 5.

The policy information is available in the `/opt/krb5/polv2` directory and the logs are available in `/tmp/kdb_migrate.log` file.

Step 5. Configure the Kerberos server v3.0.

You can configure Kerberos server manually or by using the `krbsetup` tool.

Ensure that the following values are the same in both versions of the Kerberos server:

- Realm name
- Master key name

The master key password must be identical to the one that was used in v1.0. This is applicable if you have not opted to change the password, as mentioned in step 3. If you have changed the password, use the same new password while creating the Kerberos server v3.0 database.

If you used the `-e` option to change the master key encryption type from v1.0 to v3.0 in step 3, use the same new encryption type for the master key while creating the database in v3.0.

If you did not specify the `-e` option in step 3, then the encryption type with which the v3.0 database was created must be the same as the one specified while creating the v1.0 database. For more information, type `man 4 kdc.conf` at the HP-UX prompt and see the `master_key_entry`.

The `krbsetup` interactive tool prompts for the required parameters. For more information, type `man 1M krbsetup` at the HP-UX prompt or see “Auto-Configuration of the Kerberos Server” on page 63.

Step 6. Load the new version of the dump file generated in step 3.

Use the `kdb_load` tool to load the database from the dump file, `/opt/krb5/dumpfilev3.0`:

```
# kdb_load -f /opt/krb5/dumpfilev3.0
```

Upon success, the following message appears:

```
“Load Successful”
```

The migration process of the principal information is now completed.

Consider the following points:

- The principal information is migrated from v1.0 to v3.0.
- The `/opt/krb5/polv2` file contains the policy-related information. You need to decide on the policies and add the policies to the `/opt/krb5/password.policy` file.

The policy applicable to the principal that is migrated from v1.0 to v3.0 is based on the instance name of the principals. To modify the policy, edit the principal to change the policy name field to the new policy.

- You cannot migrate the `admin_acl_file`. You need to add the appropriate ACLs to the `/opt/krb5/admin_acl_file` using the old `admin_acl_file`. For more information, see "The admin_acl_file File" on page 113.
- The `/tmp/kdb_migrate.log` file contains the log messages of step 3.

  The log messages inform you of the failure ([ERR] message), successful migrations ([LOG] messages), and so forth.

  If you encounter any problem while loading the new version of the dump file, analyze the dump file.

Copy the `/etc/krb5.conf` file of the v1.0 server to the new system, where you are installing the v3.0 server. In addition, copy the `/var/adm/krb5/krb5kdc/kdc.conf` file if the master key principal name is not the default `K/M`. If only the master key principal name differs from the default, avoid copying the `kdc.conf` file by specifying the `-M` option while using the `kdb_migrate` tool.

Migrating from Kerberos Server Version 2.0 to Version 3.0

If you want to use the Kerberos server with C-tree as the backend database, migrate your existing Kerberos server to Kerberos server v3.0.

In the Kerberos server v2.x, the password policy was based on the instance name to which the principal belongs. Starting with the Kerberos server v3.0, the password policy is not based on the instance name but is based on the policy subscribed to the principal, which provides the flexibility for a principal to subscribe to any policy in the `/opt/krb5/password.policy` file.

You must securely copy the `adm_acl_file` from the Kerberos server v2.0 to the v3.0 system.

IMPORTANT

After migrating the v2.0 database to the v3.0 server, you must modify the v2.0 principals with the appropriate policy names (policy names are present in the `/opt/krb5/password.policy` file). The instance-based rules apply if you do not specify the policy name.

To retain the v2.0 policies, copy the `password.policy` file to the v3.0 server before creating a new principal.

You can change the policy name using one of the administrative tools: `kadminl`, `kadmin`, `kadminl_ui` or `kadmin_ui`.

When you migrate the v2.0 database to the v3.0 server, the default principal of the v2.0 database does not contain the policy name field. Therefore, the default policy applicable to the created principals is `*` (the default policy), until you modify the default policy of the principal.

To migrate from Kerberos server v2.0 to v3.0, complete the following steps:

Step 1. Dump the database on the v2.0 server.

On the Kerberos server v2.0, dump the database with the default dump version. The dump file must contain the default header, “`kdb5_util load_dump version 5.0`”.

```
# kdb_dump -f /opt/krb5/dumpfilev2.0
```

Step 2. Copy the dump file to the system on which you are installing the v3.0 Kerberos server

Step 3. Install the v3.0 Kerberos daemons on the new system.

Step 4. Configure the Kerberos server v3.0.

NOTE Ensure that the following values are the same on both the versions of the Kerberos server:

- Realm name
- Master key name

Ensure that the master key password is identical to the one that was used in v2.0:

```
# krbsetup
```

The instance-based policy applies if you do not subscribe principals to a specific policy.

You can configure the Kerberos server manually or by using the `krbsetup` tool. This is an interactive tool that prompts you for the required parameters. For more information, type `man 1M krbsetup` at the HP-UX prompt or see “Auto-Configuration of the Kerberos Server” on page 63.

Step 5. Load the dump file generated in step 1 using the following command:

```
#kdb_load -f <dump_filename>
```

On successful completion, the following message is displayed:

```
Load Successful
```

Now, the migration process of the principal information is completed.

Migrating from Kerberos Server Version 3.0 to Version 3.1

If you want to use the Kerberos server with LDAP as the backend database, migrate your existing Kerberos server to Kerberos server v3.0.

Use the `krb_2_ldap` utility to migrate information of the previous version of the Kerberos server to the LDAP database. The `krb_2_ldap` utility performs the following tasks, while migrating information:

- Converts each entry of the version 2.0 or 3.0 dumpfile to `ldif` file entry. The new entries are dumped into an LDIF file.
- Logs any log messages or errors and displays it in stdout format.

Complete the following steps to migrate from Kerberos server v3.0 to v3.1:

Step 1. Dump the database on the v3.0 server.

On the Kerberos server v3.0, dump the database with the default dump version. The dump file must contain the default header, "`kdb5_util load_dump version 5.0`".

```
# kdb_dump -f /opt/krb5/dumpfilev3.1
```

Step 2. Use the `krb_2_ldap` utility to create the LDIF file.

```
# krb_2_ldap -d <dump filename> -l <ldif filename>
```

Step 3. You must manually edit the `LDIF` file.

Uncomment the first two lines of the `LDIF` file. Replace the `DN` name and the `changetype`, if necessary.

Step 4. Load the `LDIF` file using the following command:

```
/opt/ldapux/bin/ldapmodify -d "cn=amathew" -w eso! -h
<hostname> -p <port number> -f <ldif filename>
```

On successful completion, the following message is displayed:

```
Load Successful
```

Now, the migration process of the principal information is completed.

4 Interoperability with Windows 2000

When you configure interoperability between the Kerberos server and the Windows 2000 operating system, you must set certain configuration

parameters. This chapter discusses what you need to know about configuring such an environment. This chapter contains information specific to establishing interoperability with Windows 2000 Kerberos implementations. Before reading this chapter, ensure that you are familiar with the concepts in Chapter 10, “Managing Multiple Realms,” on page 275.

This chapter discusses the following topics:

- “Understanding the Terminology” on page 53
- “Kerberos Server and Windows 2000 Interoperability” on page 55
- “Establishing Trust Between Kerberos Server and Windows 2000” on page 56
- “Single Realm (Domain) Authentication” on page 58
- “Interrealm (Interdomain) Authentication” on page 59
- “Special Considerations for Interoperability” on page 60

Understanding the Terminology

Both the Kerberos server and Microsoft® provide Kerberos security for your network. While the technology is the same, the terminology varies.

Kerberos authentication depends upon establishing trust between users and services through a trusted third party called a Key Distribution Center (KDC). HP provides a KDC on the security server, and Windows 2000 provides a KDC on the domain controller.

Each KDC stores information about trusted users and services in a central database called the principal database in HP terms and the Active Directory of the domain in Microsoft terms. Each database contains a collection of users. In HP terms, the database contains a collection called a realm and each entry is called a principal. In Microsoft terms, the database contains a collection called a domain and each entry is called an account.

The most important information associated with any principal in the Kerberos model is its unique symmetric key, that is, the key used to encrypt and decrypt information on behalf of the principal. HP uses the term, secret key; Microsoft uses the terms long-term key or shared principal key. The KDC, as the trusted third party, shares a unique secret key with all of its principals. When a principal and the KDC exchange information to establish trust, the principal uses its secret key to encrypt the message. The KDC decrypts the message using the secret key of the principal stored in the database and then attempts to authenticate the principal.

During logon, if KDC successfully authenticates the user, it responds with a special message, called a ticket granting ticket (TGT). The ticket entitles you to request access to other services known to the KDC.

The client system stores the ticket in memory. In HP terminology, the client system stores the ticket in the credentials cache and uses it to request service tickets to authenticate the applications or services on the network. In Microsoft terminology, the client system stores the ticket in the secure cache and uses it to request session tickets to authenticate to applications or services.

The HP and Microsoft implementations of Kerberos have virtually identical conceptual frameworks, but mechanical differences exist. For example, the HP implementation uses configuration files to locate host

systems and the Microsoft implementation uses a DNS lookup to resolve host names. But both implementations are written to RFC 1510 (*The Kerberos Network Authentication Service (V5)*) and RFC 1964 (*The Kerberos Version 5 GSS-API Mechanism*), and hence they can interoperate.

Table 4-1 summarizes analogous terminology in the Kerberos server and Windows 2000 Kerberos implementations.

Table 4-1 Table of Analogous Terms

Kerberos Server	Windows 2000
Realm	Domain
Interrealm	Interdomain Crossdomain Crossrealm
Secret key	Longterm key Shared principal key
Credentials cache	Secure cache
Principal database	Active directory
Service ticket	Session ticket
Security server	Domain controller
Principal names	Account names

Kerberos Server and Windows 2000 Interoperability

Following are the possible interrealm interoperability scenarios between the Kerberos server software and Windows 2000, each with its own configuration requirements.

Scenario 1

A Windows 2000 user needs access to services in a Kerberos server realm. Here, the Kerberos server realm is the target realm and the Windows 2000 domain is the source realm. The Kerberos server must trust the Windows 2000 domain controller to perform secure authentication.

Scenario 2

A Kerberos server principal needs access to services in a Windows 2000 domain. Here, the Windows 2000 domain is the target realm and the Kerberos server realm is the source realm. The Windows 2000 domain controller must trust the Kerberos server to perform secure authentication.

Scenario 3

The Kerberos server principals and Windows 2000 users must access services in the realm or domain. Two-way trust must exist between the Kerberos server and the Windows 2000 domain controller.

Establishing Trust Between Kerberos Server and Windows 2000

To establish trust between Kerberos server `KRB.REALM` and Windows 2000 `W2K.DOMAIN`, complete the following steps:

Step 1. Add interrealm service principals to the Kerberos server realm. For more information, see “HP Kerberos Administrator” on page 132.

- If the realm is the source realm, the name of the principal is `krbtgt/W2K.DOMAIN@KRB.REALM`.
- If the realm is the target realm, the name of the principal is `krbtgt/KRB.REALM@W2K.DOMAIN`.

Step 2. On the Windows 2000 domain controller, use the Active Directory Domains and Trusts snap-in to create the trust relationship.

- If the domain trusts the Kerberos server realm, add the realm name to the Domains that this domain trusts field.
- If the Kerberos server realm trusts the Windows 2000 domain, add the realm name to the Domains that trust this domain’ field. Keep in mind that the passwords in steps 1 and 2 must be identical for the corresponding principals.

Step 3. Update the client configuration files or the DNS configuration with the name of the foreign KDC.

- For the Kerberos server clients, perform the following steps:
  a. Add the Windows 2000 domain controller domain name and fully qualified domain name to the `/etc/krb5.conf` file of the client.
  b. Configure the `[capaths]` section for the direct trust relationship between the realms.
  c. Add the host-to-realm name mapping data for each available Windows 2000 service to the `/etc/krb5.conf` file of the client.
- To invoke the Windows 2000 `Ksetup` tool on the Windows 2000 client, execute the following command:

```
Ksetup/addkdc KRB.REALM <fqdn>
```

NOTE The `fqdn` qualifier specifies the fully qualified domain name of the Kerberos KDC.

Step 4. Reboot the Windows 2000 domain controller. You need not reboot the Kerberos server or client.

Single Realm (Domain) Authentication

Single realm interoperability scenarios involve one or more client systems in a given realm or domain that authenticate to a single KDC. Following are the interoperability scenarios that do not require interrealm authentication:

- Kerberos server principals and Windows 2000 users can authenticate to a Kerberos server and access services registered in that realm.
- Kerberos server principals and Windows 2000 users can authenticate to a Windows 2000 domain controller and access services registered in that domain.

Single realm authentication requires all Kerberos server principals and Windows 2000 users to be entered in the same database regardless of whether that is a principal database on a Kerberos server or a Windows 2000 domain controller.

IMPORTANT In single realm authentication, principals can only access resources in their native realm. If a principal needs access to resources in a different realm, you must configure interrealm authentication.

Interrealm (Interdomain) Authentication

If two distinct realms share common keys, the realms trust one another. With that trust in place, principals can securely access services in their native realm as well as those in the trusted realm. HP calls such an access interrealm authentication, and Microsoft calls it inter-domain authentication or cross-realm authentication.

The following are examples of interrealm interoperability scenarios:

- A Kerberos principal can authenticate to a Kerberos server with access services registered in its native realm and trusted Windows 2000 domains.
- A Kerberos principal can authenticate to a Windows 2000 domain controller with access services registered in its native domain and in trusted foreign domains or realms.
- A Windows 2000 principal can authenticate to a Kerberos server with access services registered in its native realm and in trusted foreign realms or domains.
- A Windows 2000 principal can authenticate to a Windows 2000 KDC with access services registered in its native domain and in trusted foreign domains or realms.

Interrealm authentication relies on secure authentication between users and the KDC in a single realm. The shared interrealm key between trusted KDCs provides the extra link to create a chain of trust that allows a principal in one realm to authenticate to a service in a trusted foreign realm.

Special Considerations for Interoperability

You must consider the following issues related to interoperability with Windows 2000 implementations.

Database Considerations

Your network can contain more than one server, but only one master copy of the database is propagated to all secondary security servers. In a Windows 2000 Kerberos implementation, an enterprise can contain more than one domain controller, and each domain controller contains a writable copy of the database. Therefore, the two Kerberos implementations cannot share the same database.

You cannot propagate database entries between Kerberos servers and Windows 2000 domain controllers. Do not attempt to set a Windows 2000 domain controller as a secondary security server to a Kerberos primary security server, or vice versa.

Encryption Considerations

In the Kerberos authentication protocol, critical information is never sent in clear text over the network. Instead, the information is encrypted using a specified algorithm. Although the Kerberos server supports 3DES encryption, Windows 2000 requires DES encryption when it interoperates with other Kerberos implementations. Thus, principals in these realms that want to access resources in Windows 2000 domains must use a DES key type.

Postdated Tickets

The Kerberos server and client supports postdated tickets, but the Windows 2000 domain controller and client do not. If you use postdated tickets to run batch procedures over time, be sure the procedure does not need access to Windows 2000 services.

5 Configuring the Kerberos Server With C-Tree Backend

This chapter describes the configuration files and procedures used to configure the Kerberos Server with C-tree backend.

Configuration Files for the Kerberos Server

You must install all the critical Kerberos server files on the system before you start configuring the Kerberos Server. You must configure these files on the primary security server and copy these files to all the secondary security servers on the network. Table 5-1 briefly describes the server files that you need to configure.

Table 5-1 Security Server Files That Require Configuration

Configuration File	Function
`/opt/krb5/krb.conf`	Describes the default realm of the primary security server and the roles of each server for that particular realm.
`/opt/krb5/krb.realms`	Provides a way to map the host name or domain name to the associated realm name.
`/opt/krb5/admin_acl_file`	Controls the administrative permissions for administrators. See “The admin_acl_file File” on page 113 for more information.
`/opt/krb5/password.policy`	Controls password policy for the entire security network. See “Password Policy File” on page 119 for more information.
`/opt/krb5/kpropd.ini`	Contains the configuration information that is used for propagation. This is a text file. See “The kpropd.ini File” on page 251 for more information.

This chapter contains detailed descriptions of the `krb.conf` and `krb.realms` configuration files. If you have opted to configure LDAP as the backend, see “Planning Your LDAP Configuration” on page 83.

The krb.conf File

The krb.conf configuration file contains information about the default realm of the host, the administration server, and security servers for known realms. HP recommends that you copy the krb.conf.sample file from the /opt/krb5/examples directory to the /opt/krb5 directory.

This file must reside in the /opt/krb5 directory and must have the following permissions:

```
-rw-r--r--      root       3
```

The configuration file identifies the servers that support authentication for the designated realm, and defines the default realm for the host where the file is stored.

The krb.conf file lists the default realm of the host system. It also maps known realms to their primary and secondary security servers by host name, and network location.

Assuming that your network environment performs load-balancing and redundancy, you must create multiple versions of the krb.conf file. You must also configure secondary security servers to act as authentication servers. This allows the primary security server to be available for tasks other than authentication.

The krb.conf file is used during propagation configuration. The realm specified in the first line of the configuration file is considered as the default realm of the server. This has to be the first realm created in the database containing the K/M principal.

The krb.conf File Format

Use the format shown below to create an entry in the krb.conf file. See Appendix B, “Sample krb.conf File,” on page 315 to see how a sample krb.conf file looks.

```
Your_Realm_Name
Your_Realm_Name Your_Secondary_Server1
Your_Realm_Name Your_Secondary_Server2
Your_Realm_Name your_primary_server admin server
```

The first line of the krb.conf file identifies the host system’s default realm. By convention, realm names are in uppercase letters to visually distinguish them from domain names.

NOTE Realm names are case sensitive; you must type the realm name correctly if your site does not follow the uppercase convention.

The subsequent lines require fields that identify the security server host names. Each field in the line must be separated by a space or a tab. The second field indicates the Fully Qualified Domain Name (FQDN) of the host security server for that realm.

The order of entries in the `krb.conf` file is important on the client system, because it is used to identify the intended order of redundant security servers. Applications attempting to connect to the security server use this file to read the entries in the listed order. Redundant security servers are used when higher priority security servers are unavailable or when a network timeout has occurred.

To create comments, use the hash sign (`#`). Ignore blank lines, leading or trailing white spaces in a line, and characters after a hash (#) symbol.

The `krb.realms` File

The `krb.realms` file defines host-to-realm or domain-to-realm name mapping data. The `krb.realms` file is located only on Kerberos server systems in the `/opt/krb5` directory.

The `krb.realms` file ensures that all systems on the network can identify the other systems that reside in each realm.

Because, the realm name is case sensitive, the Kerberos Server looks for a domain name that is in uppercase characters. If you decide to follow the default realm naming convention, the realm names are already in uppercase characters, and you need not configure and maintain the `krb.realms` file on your client system.

Secure applications initially search for a matching host name and then a matching domain name in the `krb.realms` file. If a match is not found, the application initiates a wildcard match.

If no translation entry applies or the file does not exist, the realm name of the host is considered as the domain name of the host's domain. This domain name is converted to the uppercase equivalent.

The krb.realms file must contain sufficient entries to define the realm used by every service a client computer must access. You can create a krb.realms file that contains all the required entries for your enterprise.

If you support inter-realm authentication, the krb.realms file must contain the required entries to locate the foreign realms.

NOTE The krb.realms file does not identify systems as primary or secondary security servers. It does not define the relationship between the primary and secondary security servers. These definitions exist in the krb.conf configuration file.

The krb.realms File Format

Use the format below to add entries in the krb.realms file. See Appendix C, “Sample krb.realms File,” on page 319to see how a sample krb.realms file looks.

```
Your_Primary_Security_Server Your_Realm_Name
.Your_Secondary_Security_Server Your_Realm_Name
*.Your_Domain_Name Your_Realm_Name
```

You can add entries to the file to identify various translations from host names to realm names. The order of the entries is insignificant.

Each entry in the file requires two fields that are separated either by a space or by a tab. The following format is generally used:

- The first field specifies a name. You can either specify a single host name or specify multiple host names with one entry using the wildcards . (period) or * (asterisk), respectively, as described in Table 5-2.
- The second field specifies the associated realm. By convention, realm names must be in uppercase letters to visually distinguish realm names from domain names.

NOTE Realm names are case sensitive. You must type the correct case for the realm name if you are not following the uppercase convention.

To create comments, use the hash sign (#). Any characters after a # sign are ignored. Blank lines and any leading or trailing white spaces in a line are also ignored.

To identify multiple hosts that belong to the same realm in a single entry, use one of the wildcard characters described in Table 5-2.

Table 5-2 Wildcard Characters

Wildcard Character	Description
. (period)	Begin the name field with a period followed by a domain name to designate that all hosts in the specified domain belong to the indicated realm. For example, to indicate that the hosts `sales.bambi.com` and `mrkt.bambi.com` belong to `REALM1`, add the following entry in your `krb.realms` file: `.bambi.com REALM1`
* (asterisk)	Begin the name field with an asterisk (*) followed by a parent domain name to designate all hosts in subdomains that belong to the indicated realm. For example, to indicate that hosts `bob.sales.bambi.com` and `man.john.sales.bambi.com` belong to `REALM2`, add the following entry in your `krb.realms` file: `*.sales.com REALM2`

Autoconfiguring the Kerberos Server

An automated tool named `krbsetup` is provided to autoconfigure your Kerberos server. Use this tool to:

- Configure the Kerberos Server with either LDAP or C-Tree as the backend database.
- Unconfigure the server.
- Start the `kdcd` and the `kadmind` daemons.

NOTE You must start the `kpropd` daemon manually if you have opted for C-Tree as the backend database.

- Stop the `kdcd`, `kadmind`, and `kpropd` daemons.

The `krbsetup` tool is installed in the following directory:

`/opt/krb5/sbin`

This tool automatically creates the following files and places them in the `/opt/krb5` directory:

- `krb.conf`
- `krb.realms`
- `krb5_ldap.conf`
- `krb5_schema.conf`
- `krb5_map.conf`

This tool allows you to:

- Specify whether you want to configure your Kerberos server with either LDAP or C-Tree as the backend database.
- Specify whether you want to configure your Kerberos server as either a primary security server or a secondary security server.
- Customize your realm name.
- Provide an option to create a stash file.

- Specify the encryption type.
- Specify a different location for the log messages if you do not want to store the log messages in the default `syslog` file.
- Specify the security mechanism for your LDAP-based Kerberos server.
- Specify the Directory server host name of the LDAP-based Kerberos server.
- Specify the TCP port number of the LDAP-based Kerberos server.
- Specify the Proxy user DN of your LDAP-based Kerberos server.
- Extend your Kerberos schema on the Directory server.
- Specify the Default base DN for search of the LDAP-based Kerberos server.
- Specify the default principal subtree DN of the LDAP-based Kerberos server.
- Specify the object class template of the LDAP-based Kerberos server.

The other sections in the configuration files are set to the default values. If you want to customize these sections, manually edit the configuration files and restart the `kdcd` and `kadmind` daemons using this tool.

NOTE HP recommends that you use the `krbsetup` tool to configure your basic Kerberos server.

Following steps show you how to autoconfigure your Kerberos server:

Step 1. Run the `/opt/krb5/sbin/krbsetup` utility.

Step 2. Select one of the following options:

```
1) Configure the server
2) Start the Kerberos daemons
3) Stop the Kerberos daemons
4) Un-configure the Server
5) Exit

6) Help
```

Step 3. To configure the server, select option **1**.

- To configure your Kerberos Server with C-Tree, select option **1**. See “Configuring the Kerberos Server with C-Tree” on page 71 to continue configuring your Kerberos Server with C-Tree.
- To configure your Kerberos Server with LDAP, select option **2**. See “Configuring the Kerberos Server with LDAP” on page 88 to continue configuring your Kerberos Server with LDAP.
- To return to the main menu, select option **0**.

Step 4. To start the Kerberos daemons, `kadmind` and `kdcd`, select option **2**. You must manually start the `kpropd` daemon. Press **Return** to return to the main menu.

Step 5. To stop the Kerberos daemons, select option **3**. Press **Return** to return to the main menu.

Step 6. To unconfigure the Kerberos daemons, select option **4**. You are prompted with a message to confirm this action. Press **`y`** to unconfigure the Kerberos server and **`n`** to return to the main menu.

Step 7. To exit the tool, select option **5**.

Step 8. To view the help contents, select option **6**.

Configuring the Kerberos Server with C-Tree

Complete the following procedure to autoconfigure your Kerberos server with C-Tree:

Step 1. Run the `/opt/krb5/sbin/krbsetup` utility.

Step 2. Select one of the following options:

```
1) Configure the server
2) Start the Kerberos daemons
3) Stop the Kerberos daemons
4) Un-configure the Server
5) Exit

6) Help
```

Step 3. To configure the Kerberos Server, select option **1**.

Step 4. To configure the Kerberos Server with C-Tree backend, select option **1**.

Step 5. To remove the existing Kerberos server configuration, press **y** and press **n** to retain the existing database.

Step 6. Configure your Kerberos server as either a primary security server or a secondary security server:

1. To configure your Kerberos server as a primary security server, select option **1**.
2. To configure your Kerberos server as a secondary security server, select option **2**. Before you log on to the Remote Administrator, stop the daemons that are already running on the secondary security server.

Step 7. Specify the encryption type. If you do not specify a value, the default value, DES-MD5, is selected.

Step 8. To stash the principal database key file on your local disk, press **y** at the prompt. Press **n** if you do not want to stash the principal database key file.

Step 9. Enter names for other servers:

- If you had chosen to configure your primary security server, you are prompted for the names of your secondary security servers.
- If you had chosen to configure your secondary security server, you are prompted for the name of your primary security server.

Step 10. Enter the realm name. The default value is displayed. To use the default, press **Return**; otherwise, enter your realm name.

Step 11. Enter the location where you want to store log messages. By default, log messages are stored in the `syslog` file. To change the default location, enter **y** and specify the absolute directory name for the log messages.

Step 12. Enter the database master password.

Step 13. Re-enter the database master password to verify the password.

Step 14. Your configuration is now complete and your Kerberos daemons are up and running. To return to the main menu, press **Return**.

6 Configuring the Kerberos Server with LDAP

This chapter describes the configuration files and procedures used to configure the Kerberos Server with LDAP backend.

Configuration Files for LDAP Integration

You must configure the LDAP configuration files listed in Table 6-1, before setting up your Kerberos server. This chapter contains detailed descriptions of these configuration files.

The `krbsetup` autoconfiguration tool generates these files, based on your input. Alternatively, you can manually edit the sample configuration files available in the `/opt/krb5/examples` directory. HP recommends that you use the autoconfiguration tool to generate these files.

Table 6-1 LDAP Configuration Files

File	Function
`krb5_ldap.conf`	Contains the LDAP configuration parameters and values. This file is used by the Kerberos server to connect to the Directory server.
`krb5_schema.conf`	Describes the object and attribute definitions that define the structure of the kerberos principal entries in the LDAP database.
`krb5_map.conf`	Defines the mapping from the default kerberos attributes to the user defined attributes.

The krb5_ldap.conf File

The `krb5_ldap.conf` file is the primary configuration file. It contains information about the LDAP configuration parameters and values for the Kerberos server.

If the `krb5_ldap.conf` file is not present in the `/opt/krb5` directory, then the Kerberos Server assumes that C-tree is the backend database.

This file is generated automatically based on the input provided by you while autoconfiguring the Kerberos server. Alternatively, a sample file is available in the /opt/krb5/examples directory. You can copy this file to the /opt/krb5 directory, and manually edit it. HP recommends that you use the autoconfiguration tool to generate this file.

This file must reside in the /opt/krb5 directory and must have the following permissions:

```
-rw-------      root       3 sys
```

The krb5_ldap.conf File Format

Following is the format of the krb5_ldap.conf file:

```
ldap_enabled = 1
directory_servers = fox.bambi.com:389
base_dn_for_search = o=bambi.com
security_mech = password
proxy_user=cn = Directory Manager
proxy_user_password = <#$%^&*0#$0^&@1!$^%#10^0%>
default_object_template = account
default_princ_subtree = ou=People,o=bambi.com
default_objcls_attr = uid
```

Use the krb5_encrypt tool to modify the proxy_user_password field in the /opt/krb5/krb5_ldap.conf file. You must change the proxy field whenever you change the password of the proxy user or the master key. Ensure that the encryption key type and the master key type are the same; else the Kerberos server will not connect to the LDAP server. Table 6-2 provides a detailed description of the various parameters in the krb5_ldap.conf file.

Table 6-2 krb5_ldap.conf File Format

Parameter	Description
ldap_enabled	This line indicates whether you have enabled LDAP. 1 indicates that you have enabled LDAP and 0 indicates that you have not enabled LDAP as the backend database.

Table 6-2 **`krb5_ldap.conf` File Format (Continued)**

Parameter	Description
`directory_server`	This line indicates a space separated list of LDAP Servers. Example: `fox.bambi.com:389 deer.bambi.com`
`base_dn_for_search`	This line indicates the default base DN for search is the root of the directory tree on the Directory server, where the Kerberos server searches for kerberos principals. Example: `ou=People, o=bambi.com`
`default_princ_subtree`	The default principal subtree DN is where all Kerberos principals are added by default, if no LDAP entry is specified while creating the kerberos principal. The default principal subtree DN must be located under the default base DN for search functionality. Example: `ou=people, o=bambi.com`
`security_mech`	This line specifies the security mechanism used to connect to the LDAP server. Currently, the supported mechanisms are Password and Secure Sockets Layer (SSL).
`default_object_template`	This line specifies the structural class, which is added by default. Example: `posixaccount`

Table 6-2 **krb5_ldap.conf File Format (Continued)**

Parameter	Description
default_objcls_attr	This line specifies the mandatory attribute of the default object class. Example: uid When the Kerberos server creates a default object it uses the first attribute specified in this field, as the naming attribute. When adding a principal, an error message is displayed if duplicate entries are found. You can change the default settings of the naming attribute by changing the order of entries in the krb5_ldap.conf file. Save these changes and restart the Kerberos server application.
proxy_user	This line specifies the DN of the proxy user. The Kerberos server binds to the Directory server as the proxy user. The proxy user must have the appropriate privileges to create, modify and delete Kerberos principals. Example: cn=Anne

The krb5_schema.conf File

A schema is a collection of object and attribute definitions that defines the structure of the entries in a database. The krb5_schema.conf file is the kerberos schema file that contains the object and attribute definitions of the kerberos principal entries. LDAP objects are standardized in order to provide interoperability with a variety of directory services servers. The krb5_schema.conf file defines the following:

- Type of object classes
- Attributes of the object classes
- Optional attributes
- Syntax of each attribute

For example, a schema can define a person object class. The person schema might require that a person have a surname attribute that is a character string. It also specifies that a person entry can optionally have a `telephoneNumber` attribute that is a string of numbers with spaces and hyphens.

The `krb5_schema.conf` file is automatically generated based on the input provided by you while autoconfiguring the Kerberos server. Alternatively, a sample file is available in the `/opt/krb5/examples` directory. You can copy this file to the `/opt/krb5` directory, and manually edit it. HP recommends that you use the autoconfiguration tool to generate this file.

This file must reside in the `/opt/krb5` directory and must have the following permissions:

```
-rw-r--r--      root       3
```

The `krb5_schema.conf` File Format

Following is the format of the `krb5_schema.conf` file:

```
dn: cn=schema
changetype: modify
add: attributetypes
attributetypes: ( hpKrbPrincipalName-oid
NAME 'hpKrbPrincipalName'
DESC 'Kerberos principal identity for a user in the form
<principal>@<realm>'
EQUALITY caseExactMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE )
attributetype: ( hpKrbMaxTicketAge-oid
NAME 'hpKrbMaxTicketAge'
DESC 'Value defining the maximum lifetime of a user ticket'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbMaxRenewAge-oid
NAME 'hpKrbMaxRenewAge'
DESC 'Value defining the maximum renewable lifetime of a
```

```
ticket'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbAccountExpires-oid
NAME 'hpKrbAccountExpires'
DESC 'Value used to compute date and time when account will
expire'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbPasswordExpireTime-oid
NAME 'hpKrbPasswordExpireTime'
DESC 'A value indicating the date and time when the password
expires'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbPwdLastSet-oid
NAME 'hpKrbPwdLastSet'
DESC 'A value that stores the date and time when the password
was last set'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbLastLogon-oid
NAME 'hpKrbLastLogon'
DESC 'A value used to compute date and time of last
successfullogon'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbBadPasswordTime-oid
NAME 'hpKrbBadPasswordTime'
DESC 'Value used to compute date and time of last unsuccessfu
logon attempt'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbBadPwdCount-oid
NAME 'hpKrbBadPwdCount'
DESC 'Number of unsuccessful attempts to authenticate with this
account'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbModifiersName-oid
NAME 'hpKrbModifiersName'
DESC 'The last modifier of any attribute associated with a
principal entry'
EQUALITY caseExactMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE )
```

```
attributetypes: ( hpKrbModifyTimestamp-oid
NAME 'hpKrbModifyTimestamp'
DESC 'The date and time when the identity specified in the
hpKrbModifiersName attribute made the last modification'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbAttributes-oid
NAME 'hpKrbAttributes'
DESC 'A value containing one or more flags'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbPolicyName-oid
NAME 'hpKrbPolicyName'
DESC 'The Kerberos password policy to which this principal
subscribes to'
EQUALITY caseExactMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE )
attributetypes: ( hpKrbKeyVersion-oid
NAME 'hpkrbAuthzData'
DESC 'Other Authorization Data.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.40
SINGLE-VALUE )
add: objectClasses
objectClasses: ( hpKrbPrincipal-oid
NAME 'hpKrbKeyVersion'
DESC 'Version of a secret key; a monotomic increasing number
beginning with 1'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
attributetypes: ( hpKrbKeyData-oid
NAME 'hpKrbKeyData'
DESC 'A set of values with each value containing an encrypted
key and information about the encrypted key.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
attributetypes: ( hpkrbAuthzData-oid
NAME 'hpKrbPrincipal'
DESC 'An auxiliary class for use in configuring an entry to
represent a Kerberos principal.'
SUP top Auxiliary MAY ( hpKrbPrincipalName $ hpKrbMaxTicketAge
$ hpKrbMaxRenewAge $ hpKrbAccountExpires $
hpKrbPasswordExpireTime $ hpKrbPwdLastSet $ hpKrbLastLogon $
hpKrbBadPasswordTime $ hpKrbBadPwdCount $ hpKrbModifiersName $
hpKrbModifyTimestamp $ hpKrbAttributes $ hpKrbPolicyName $
hpkrbAuthzData) )
```

```
objectClasses: ( hpKrbKey-oid
NAME 'hpKrbKey'
DESC 'An structural object class used for configuring the
principal name of an associated principal entry.' SUP top
STRUCTURAL MUST ( hpKrbPrincipalName ) MAY ( hpKrbKeyVersion $
hpKrbKeyData ) )
```

The `krb5_map.conf` File

The `krb5_map.conf` mapping file defines the mapping of the default kerberos attributes to user defined attributes, to support the Kerberos server schema. The Kerberos server uses this map file for translating Kerberos attribute names to LDAP attribute names. Each entry in the mapping file represents a translation for an attribute.

The `krb5_map.conf` file is automatically generated based on the input provided by you while autoconfiguring the Kerberos server. Alternatively, a sample file is available in the `/opt/krb5/examples` directory. You can copy this file to the `/opt/krb5` directory, and manually edit it. HP recommends that you use the autoconfiguration tool to generate this file.

This file must reside in the `/opt/krb5` directory and must have the following permissions:

```
-rw-r--r--      root       3
```

The `krb5_map.conf` File Format

Following is the format of the default mapping file:

```
hpKrbPrincipalName          =  hpKrbPrincipalName
hpKrbMaxTicketAge           =  hpKrbMaxTicketAge
hpKrbMaxRenewAge            =  hpKrbMaxRenewAge
hpKrbAccountExpires         =  hpKrbAccountExpires
hpKrbPasswordExpireTime     =  hpKrbPasswordExpireTime
hpKrbPwdLastSet             =  hpKrbPwdLastSet
hpKrbLastLogon              =  hpKrbLastLogon
hpKrbBadPasswordTime        =  hpKrbBadPasswordTime
hpKrbBadPwdCount            =  hpKrbBadPwdCount
hpKrbModifiersName          =  hpKrbModifiersName
hpKrbModifyTimestamp        =  hpKrbModifyTimestamp
hpKrbAttributes             =  hpKrbAttributes
hpKrbPolicyName             =  hpKrbPolicyName
```

```
hpKrbAuthzData              =  hpKrbAuthzData
hpKrbKeyVersion             =  hpKrbKeyVersion
hpKrbKeyData                =  hpKrbKeyData
```

Planning Your LDAP Configuration

The following sections of this chapter describe how to plan and configure your Kerberos Server to work with the Directory server.

Before You Begin

Remember the following points when you plan your LDAP setup:

- Use the Configuration Worksheet (see Appendix A, “Configuration Worksheet,” on page 311) to record your decisions and other information that you need later for configuration.
- You must have an LDAP directory. You can obtain the Netscape Directory server 6.0 (J4258CA) for HP-UX from your local HP sales office or http://www.hp.com and view the documentation at http://docs.hp.com/hpux/internet/#Netscape%20Directory%20Server.
- See the white paper Preparing Your Directory for HP-UX Integration at http://docs.hp.com/hpux/internet for advice on how to set up and configure your directory to work with HP-UX.
- Most examples here use the Netscape Directory server 6.0 (J4258CA) for HP-UX and assume you have some working knowledge of this directory and its tools, such as the Directory Console and ldapsearch. If you have another directory, consult your directory’s documentation for specific information.
- The examples use a base DN of `o=bambi.com` for illustrative purposes.

Setting up Your LDAP Configuration

Plan how to set up and verify your LDAP directory and your Kerberos server environment, before you put them into production. Consider the following questions and record your decisions and other information that you will need later in the Configuration Worksheet found in Appendix A, “Configuration Worksheet,” on page 311.

- What is the host name of your directory server?

  Write down your directory server host name in the Configuration Worksheet. This is where your Kerberos principals reside. Enter either the FQDN or the IP address.

  For example, `fox.bambi.com` or `18.13.118.130`.

- What is the port number of your directory server?

  Write down the port number of your directory server in the Configuration Worksheet.

  — If you have opted for SSL as the security mechanism the default TCP port number is 636.

  — If you have opted for Password as the security mechanism the default TCP port number is 389.

- Have you decided to extend the schema?

  A schema is the collection of object class and attribute type definitions. A server uses these definitions to determine how to match a filter or attribute against the attributes of a specific entry and whether to grant permissions to any given attributes.

  You must have administrative privileges to extend the schema. If you do not have these privileges contact your LDAP administrator. You need to extend the LDAP schema with Kerberos specific object classes and attributes.

- Have you decided on the security mechanism?

  To access the information stored in the directory, you must authenticate to the directory first. Once authenticated, and depending on the authorization information stored in the directory

you can access the information in the directory. Hence, you need to choose an authentication method. Currently, the supported mechanisms are Password and SSL.

The SSL protocol was devised to provide both authentication and data security. SSL encapsulates the TCP/IP socket so that every TCP/IP application can use it to secure its communication. This enables clients to verify the identity of the server and to encrypt communication of the basic authentication from the clients to the server on insecure networks. To ensure message integrity and privacy, SSL has the following features:

— Provides a hashing algorithm

— Provides for the creation and use of an encrypted communication channel

If you choose Password as the security mechanism then the client authenticates to an LDAP server by sending a bind request to the server.

NOTE In the Password security mechanism, passwords are transmitted in clear text and are vulnerable to snooping.

The primary advantage of using Password is that it is the required authentication method as defined in the LDAP standard, and all directory servers support it.

- What is the name of your default base `DN` for search?

  Entries are organized in a tree-like structure called the Directory Information Tree (DIT). Entries are arranged within the DIT based on their DNs. Distinguished Name (DN) is a unique name that unambiguously identifies a single entry. DNs are made up of a sequence of Relative Distinguished Names (RDNs). Each RDN in a DN corresponds to a branch in the DIT leading from the root of the DIT to the directory entry. A DN is composed of a sequence of RDNs separated by commas.

  For example, `ou=people`, `o=bambi.com`

  The default base DN for search is the root of the directory tree on the Directory server, where the Kerberos server searches for kerberos principals.

- What is the name of your default principal subtree `DN`?

  Each RDN in a DN corresponds to a branch in the DIT leading from the root of the DIT to the directory entry. The search base node subtree designates all the containers for the various information types under the base DN.

  For example, `ou=accounts`, `ou=people`, `o=bambi.com`

  By default, all Kerberos principals are added in the default principal subtree, if no LDAP entry is specified while creating the kerberos principal. The default principal subtree DN must be located under the default base DN for search.

NOTE To effectively search for data you must add all subtree entries under the default base DN.

- Where are your certificates located?

  This path defines the location of the database that contains the certificates for your client. The database must contain the `cert7.db` certificate, which is used by Mozilla or Netscape client.x. You must specify the path to the directory containing the certificate database.

  For example, `/.netscape/cert7.db`.

- What is the name of your proxy user?

  Write down the distinguished name of the proxy user, if needed. The Kerberos server binds to the Directory server as the proxy user. This user must have the appropriate privileges to create, modify and delete Kerberos principals.

  For example, `cn=Anne`.

- What is the name of your default object class template?

  The Kerberos principal must be associated with at least one structural object class on the Directory server. The Kerberos server uses this template for those Kerberos principals who do not have an existing object class to be associated with on the Directory server.

  For example, `posixaccount`.

- What are the attributes of your object class?

This line specifies the mandatory attributes of the default object class.The object class attribute determines the attributes the entry must have and can have. When the Kerberos server creates a default object it uses the first attribute specified in this field, as the naming attribute.

For example, `uid. cn`, `homedirectory`, `gidnumber`, `uidnumber`.

Autoconfiguring the Kerberos Server With LDAP Integration

An automated tool named `krbsetup` is provided to autoconfigure your Kerberos server. For more information on the `krbsetup` tool, see “Autoconfiguring the Kerberos Server” on page 69.

Configuring the Kerberos Server with LDAP

Complete the following procedure to autoconfigure your Kerberos server with LDAP:

Step 1. Run the `/opt/krb5/sbin/krbsetup` utility.

Step 2. Select one of the following options:

```
1) Configure the server
2) Start the Kerberos daemons
3) Stop the Kerberos daemons
4) Un-configure the Server
5) Exit

6) Help
```

Step 3. To configure the Kerberos Server, select option **1**.

Step 4. To configure the Kerberos Server with LDAP backend, select option **1**.

Step 5. To remove the existing Kerberos server configuration, press **y** and press **n** to retain the existing database.

NOTE Ensure that you have a dump of the existing Kerberos database, before you configure the Kerberos server with LDAP. “Migrating to a Newer Version of the Kerberos Server” on page 41, for more information.

Step 6. Select one of the following options to configure the security mechanism of your LDAP-based Kerberos server:

1. SSL
2. Password

Step 7. Enter the host name of the directory server. The default value is displayed. To use the default, press **Return**; otherwise, enter your fully qualified host name or the IP address.

Step 8. Enter the port number of the directory server. If you do not specify any value the following default values are selected:

- If you have opted for SSL as the security mechanism the default value 636 is selected.
- If you have opted for Password as the security mechanism the default value 389 is selected.

Step 9. Enter the DN of the proxy user. The default value is displayed. To use the default, press **Return**.

NOTE The proxy user must have the privileges to add, modify, and delete Kerberos information on the Directory server.

Step 10. Enter the Proxy User password.

Step 11. Enter the Certificate db path, if you have opted to configure SSL as the security mechanism of your LDAP-based Kerberos server.

Step 12. To extend the existing schema in the directory, press **y**. Press **n** if you do not want to extend the schema.

NOTE You must have administrative privileges to extend the schema. Contact your LDAP administrator if you do not have these privileges.

If you have pressed **y**, that is, opted to extend the schema, you are prompted for the following input:

a. Enter the DN of the Admin user. The default value is displayed. To use the default, press **Return**; otherwise, enter your DN name.

b. Enter the password.

c. Select the following object classes to remap the attributes:

1. `hpKrbPrincipal`

2. `hpKrbKey`

To remap the attributes of the object class `hpKrbPrincipal`, select option **1**.

To remap the attributes of the object class `hpKrbKey`, select option **2**.

NOTE HP recommends that you use the default attributes of the `hpKrbPrincipal` and `hpKrbKey` object classes.

Step 13. Enter the default base DN for search. The default value is displayed. To use the default, press **Return**.

Step 14. Enter the default principal subtree DN. The default value is displayed. To use the default, press **Return**.

Step 15. Enter the default template object class. The default value is displayed. To use the default, press **Return**.

Step 16. Configure your Kerberos server as either a primary security server or a secondary security server:

1. To configure your Kerberos server as a primary security server, select option **1**.
2. To configure your Kerberos server as a secondary security server, select option **2**. Before you log on to the Remote Administrator, stop the daemons that are already running on the secondary security server.

Step 17. Specify the encryption type. If you do not specify a value, the default value, DES-MD5, is selected.

Step 18. To stash the principal database key file on your local disk, press **y** at the prompt. Press **n** if you do not want to stash the principal database key file.

Step 19. Enter names for other servers:

- If you chose to configure your primary security server, you are prompted for the names of your secondary security servers.
- If you chose to configure your secondary security server, you are prompted for the name of your primary security server.

Step 20. Enter the realm name. The default value is displayed. To use the default, press **Return**; otherwise, enter your realm name.

Step 21. Enter the location where you want to store log messages. By default, log messages are stored in the `syslog` file. To change the default location, enter **y** and specify the absolute directory name where you want to store the log messages.

Step 22. Enter the database master password.

Step 23. Re-enter the database master password to verify the password.

Step 24. Your configuration is now complete and your Kerberos daemons are up and running. To return to the main menu, press **Return**.

Manually Configuring the Kerberos Server with LDAP

This section describes how to manually configure your Kerberos server with LDAP. HP recommends that you use the autoconfiguration tool to set up your basic Kerberos security server with LDAP. For more information on autoconfiguration, see “Autoconfiguring the Kerberos Server With LDAP Integration” on page 88.

The subsequent sections describe the configuration files and the steps required to manually configure your Kerberos security server with LDAP.

Editing the Configuration Files

You can manually edit the following files to configure the Kerberos security server with LDAP:

- LDAP-based Kerberos configuration file - `krb5_ldap.conf`.
- Kerberos schema file - `krb5_schema.conf`.
- Kerberos mapping file `krb5_map.conf`.
- Kerberos configuration file – `krb.conf`.
- Kerberos realms file – `krb.realms`.

The `krb5_ldap.conf` configuration file specifies the LDAP configuration information. See “The krb5_ldap.conf File” on page 74 for more information on the configuration parameters.

NOTE

You must use the `krb5_encrypt` tool to set the value of `proxy_user_password` field. Refer the *krb5_encrypt(1m)* manpage for more information on the `krb5_encrypt` tool.

The `krb5_schema.conf` schema file is the default schema. HP recommends keeping the default schema. If you choose to extend the Kerberos schema, follow the guidelines listed below:

- Never delete any element of your Kerberos schema as this affects the compatibility of your schema to other LDAP services (servers and clients).
- Never change the Kerberos schema of your directory by modifying the existing elements as this also affects the compatibility of your schema to other LDAP services.
- Never map an existing attribute name to a kerberos attribute name. This may result in an error when configuring the schema.
- Never edit the Kerberos mapping file, `krb5_map.conf`, after configuring the server.
- If you want to modify an element in the existing schema, you must also ensure that the changes are reflected in the `krb5_map.conf` mapping file.
- If you want to manually load the Kerberos schema, use the default schema located at `/opt/krb5/examples`.
- Always save your current schema before you start this process.

The Kerberos mapping file, `krb5_map.conf`, defines the mapping of the default kerberos attributes to user defined attributes, to support the Kerberos server schema. See “The krb5_map.conf File” on page 81, for more information.

The Kerberos configuration file, `krb.conf`, specifies the security servers available for client authentication and defines the default realm for the host.

The Kerberos realms file, `krb.realms`, defines the `host-to-realm` or `domain-to-realm` mapping data.

These files are available in the `/opt/krb5/examples` directory. You can copy these files to the `/opt/krb5` directory, and manually edit them.

Modify the configuration files `/opt/krb5/krb5_ldap.conf`, `/opt/krb5/krb5_schema.conf`, and `/opt/krb5/krb5_map` to reflect the correct information.

For more information about modifying the configuration files, see “Configuring the Primary Security Server” on page 96.

7 Configuring the Primary and Secondary Security Server

This chapter describes the procedure to configure the primary and secondary security server.

Configuring the Primary Security Server

The following sections describe the initial configuration tasks you need to perform to get your primary and secondary security server up and running.

The primary security server requires the following basic configuration tasks:

1. Execute the `krb5_encrypt` command to generate the master key.

NOTE If you have opted to configure your Kerberos with LDAP as the backend, specify the master key, generated by executing the `krb5_encrypt` command, in the `proxy_user` field in the `krb5_ldap.conf` file.

2. “Create the Principal Database After Installation” on page 96
3. “Add an Administrative Principal” on page 97
4. “Create the host/<fqdn> Principal and Extracting the Service Key” on page 98
5. “Start the Kerberos Daemons” on page 99
6. “Define Secondary Security Server Network Locations” on page 100

Create the Principal Database After Installation

If you choose not to create the principal database during installation, create it before configuring the security server. To create the principal database, execute the following command:

```
kdb_create -s
```

NOTE The `kdb_create` command uses the 3DES encrypted database by default.

If you are using Kerberos server v2.0 or v3.0, and want to migrate the principal database to Kerberos server v3.1, see Chapter 3, “Migrating to a Newer Version of the Kerberos Server,” on page 41.

Add an Administrative Principal

Use the HP Kerberos Administrator (`kadminl_ui`) instead of the command-line administrator (`kadminl`) to add the principal account. For more information on using the HP Kerberos Administrator and the command-line administrator, see “The kadmin and kadminl Utilities” on page 130.

Though it is possible to use the `kadmin` option to create an administrative principal, you cannot use `kadmin` to assign administrative privileges. If you want to use the `kadmin` utilities to manage your administrative principals, use a text editor to add the required entries to the file.

NOTE You must log on as a root user, on the primary security server, to add an administrative principal.

For the first administrative principal, HP recommends that you assign all permissions, indicated by `*` in `admin_acl_file`. For more information, see “The admin_acl_file File” on page 113.

You can add an administrative principal through the HP Kerberos Administrator GUI, or through the command-line interface.

To add an Administrative Principal Using the HP Kerberos Administrator

Following steps show you how to add an administrative principal using the HP Kerberos Administrator:

Step 1. Invoke the HP Kerberos Administrator using the command `kadminl_ui`.

Step 2. Add a new principal to the default realm using the following syntax:

```
# identifier/admin@DEFAULT_REALM
```

Step 3. Assign the password.

Step 4. Use the **Edit>Edit Administrative Permissions** menu to assign ALL administrative permissions to the principal.

Step 5. On the Attributes tab, clear the **Require Password Change** checkbox to disable the password change requirement.

You can also disable the password change requirement by setting the `NoReqChangePwd` setting in the principal's password policy file to 1.

NOTE By default, the principal account requires a password change at the first logon. However, `kadmin` does not permit password changes, unless you have explicit permissions to change your password.

Step 6. Save your changes and close the HP Kerberos Administrator.

For more information on using the HP Kerberos Administrator, see "HP Kerberos Administrator" on page 132.

To Add an Administrative Principal Through the Command Line

Following steps show how to add an administrative principal through the command-line interface:

Step 1. Run the `kadmin` command-line administrator.

Step 2. Add a new principal to the default realm using the following syntax:

```
command: add
Name of Principal to add: admin
Enter password:password
Re-enter password for verification:password
Enter policy name (Press enter key to apply default policy):
Principal added
```

For more information on assigning administrative privileges to principals, see "Manual Administration Using kadmin" on page 202.

Create the host/<fqdn> Principal and Extracting the Service Key

To allow principal database propagation, the primary security server must have a `host/<fqdn>` principal and the service key for this principal must be extracted to the service key table file of the server.

The host/<fqdn> principal is not automatically added to the principal database during security server software installation; you must manually add the host/<fqdn> principal using the kadminl_ui or kadminl command.

NOTE You must log on as a root user, on the primary security server, to add the host/<fqdn> principal to the database.

HP recommends that you create a host/<fqdn> principal and extract its service key using the ktutil command. To do this, type the following command at the prompt:

```
# kadminl -R "ext host/<fqdn>"
```

The host/<fqdn> is added to the principal database, along with a random key. The random key is added to the service key table. To verify that these operations are successful, use the ktutil-k command to list the contents of the key table file. The existence of a host/entry file indicates that the principal has been successfully added to the database with a random key.

NOTE Propagation is disabled if you select LDAP as your backend database. Check with your LDAP administrator, for more information about propagation of information on the LDAP Server.

Start the Kerberos Daemons

You can use the krbsetup tool to start the following Kerberos daemons:

- kdcd
- kadmind

NOTE You cannot use the krbsetup tool to start the kpropd daemon. Start the kpropd daemon manually.

Alternatively, you can use the following command to start the Kerberos daemons `kdcd` and `kadmind`:

```
/sbin/init.d/krbsrv start
```

To start the `kpropd` daemon, use the following command:

```
/opt/krb5/sbin/krpopd
```

NOTE Propagation is disabled if you select LDAP as your backend database. Check with your LDAP administrator, for more information about propagation of information on the LDAP Server.

Define Secondary Security Server Network Locations

To configure propagation, alter the Kerberos configuration files to define server network locations. For more information, see Chapter 9, “Propagating the Kerberos Server,” on page 241.

For each secondary security server installed on your network, edit the `krb.conf` file on the primary security server by adding an entry to define the role of this secondary security server in the realm. For more information on the configuration files, see “The krb.conf File” on page 65.

Security Policies

The following files are directly related to the security of the network in your organization:

- `password policy`
- `admin_acl_file`

Password Policy File

The password policy file controls password rules, such as password length, number of character types, and the lifetime of a password. This file is located on each primary security server and on all the secondary security servers.

For more information on the Password Policy File, see “Password Policy File” on page 119.

The admin_acl_file

The `admin_acl_file` lists the various administrators and their respective administrative permissions; it also lists the principals whose attributes cannot be changed without explicit privileges. This file is located in the primary security server. Protect `admin_acl_file` with appropriate read-write privileges and assign access permissions only for users with access privileges.

For more information on the adding entries to `admin_acl_file` and assigning administrative permissions, see “The admin_acl_file File” on page 113.

Starting the Security Server

After creating the Kerberos database and setting up the administrative principals, you can start the Kerberos daemons on the primary security server. To do this, edit the `/etc/rc.config.d/krbsrv` file to reflect the following values:

```
KDC = 1
ADMD = 1
```

Then, type the following:

```
/sbin/init.d/krbsrv start
```

You can also start the kerberos daemons using the following commands:

```
# /opt/krb5/sbin/kdcd
# /opt/krb5/sbin/kadmind
```

Check the startup messages in the system log files to verify whether the daemons have started properly. These programs return the following common error message:

```
Address already in use.
```

This message indicates that you have configured `kdcd` or `kadmind` to use a port that is already being used by another program. Because KDC uses the well-known port 88, this error indicates that a previous instance of the KDC is still running.

Configuring the Secondary Security Servers with C-Tree

You can now configure the secondary security servers. Assuming that you are setting up the primary security server so that you can easily switch the primary security server with one of the secondary security servers, you must perform each of the steps on the primary security server as well as on the secondary security server.

All secondary security servers require the following basic configuration tasks:

- Creating the principal database.
- Copying the Kerberos configuration files.
- Creating a `host/<fqdn>` principal and extract its key.

Creating the Principal Database

By default, the Kerberos security server uses DES3 to encrypt the principal database. If you are using DES encryption to secure your principal database, use the following command:

```
kdb_create -s -e enctype
```

where `enctype` is `DES-CBC-CRC`, `DES-CBC-MD5`, or `DES3-CBC-MD5`. You can also specify **1** for `DES-CBC-CRC`, **3** for `DES-CBC-MD5`, and **5** for `DES3-CBC-MD5`.

Copying the Kerberos Configuration File

Each secondary security server must have a copy of the Kerberos configuration files from the primary security server. The following is the default path and file name:

```
/opt/krb5/krb.conf
```

Following lists the default configuration files required on the secondary security server:

- `krb.conf`
- `krb.realms`

Creating a host/<fqdn> Principal and Extracting the Key

To allow principal database propagation, each secondary security server must contain a `host/<fqdn>` principal. You must also extract the key for the `host/<fqdn>` principal to that service key table file of the server.

You can create a `host/<fqdn>` principal and extract its key on a secondary security server by using the same procedure that is used on the primary security server. You need not log on as a root user to perform these tasks on a secondary security server. You can run `kadmin` and log on using the administrative principal name and password when prompted. For more information, see “Create the host/<fqdn> Principal and Extracting the Service Key” on page 98.

Each KDC must have a host service principal in the Kerberos database. You can create a host service principal from any host if the `kadmind` daemon is running.

Configuring the Secondary Security Servers with LDAP

You can now configure the secondary security servers. Assuming that you are setting up the primary security server so that you can easily switch the primary security server with one of the secondary security servers, you must perform each of the steps on the primary security server as well as on the secondary security server.

All secondary security servers require the following basic configuration tasks:

- Copying the Kerberos configuration files.
- Creating a stash file using the `kdb_stash` utility.

Copying the Kerberos Configuration File

Each secondary security server must have a copy of the Kerberos configuration files (`krb.conf`)from the primary security server. The `krb.conf` file is located at:

`/opt/krb5/krb.conf`

Following lists the default configuration files required on the secondary security server:

- `krb.conf`
- `krb.realms`
- `krb5_ldap.conf`
- `krb5_schema.conf`
- `krb5_map.conf`

Creating a stash file using the `kdb_stash` utility

You must create a stash file using the `kdb_stash` utility. This utility stores the master key in a stash file that the Kerberos server accesses when the security server daemons start up. You must specify the same

key type and master password that was specified when the database was created.If you run the `kdb_create` utility with the `-s` option, a stash file is created automatically.

NOTE The `kdb_stash` utility requires super user privileges to execute.

Using Indexes to Improve Database Performance

Most LDAP servers use indexes to improve search performance. Indexes are files stored in your directory databases. Separate index files are maintained for each database in your directory. Each file is named according to the attribute it indexes. The index file for a particular attribute can contain multiple types of indexes. This feature enables you to maintain several types of indexes for each attribute. For example, a file called `sn.db3` contains all of the indexes for the surname name attribute.

NOTE To improve the performance of the directory server you must index the database with attributes of `hpKrbPrincipalName`.

Depending upon the types of applications using your directory, you will use different types of indexes.

Indexes greatly improve the performance of searches in the directory databases, but they do so at the cost of slower database modification and creation operations. Indexes also cost more in terms of system resources, especially disk space.

NOTE HP recommends that you use the `Presence` and `Equality` type of index for the attribute `hpkrbprincipalName`.

For information on how to create these indexes, consult your directory server documentation.

8 Administering the Kerberos Server

This chapter explains how to administer and maintain the Kerberos database and how to manage principals using the HP Kerberos

Administrator, a graphical user interface, or the command-line administrator.

This chapter discusses the following topics:

- "Administering the Kerberos Database" on page 111
- "The kadmind Command" on page 112
- "The admin_acl_file File" on page 113
- "Password Policy File" on page 119
- "Principals" on page 121
- "The kadmin and kadminl Utilities" on page 130
- "HP Kerberos Administrator" on page 132
  - "Local Administrator – kadminl_ui" on page 134
  - "Remote Administrator – kadmin_ui" on page 198
- "Manual Administration Using kadmin" on page 202
- "Principal Database Utilities" on page 224
- "Kerberos Database Utilities" on page 225
- "Destroying the Kerberos Database" on page 229
- "Dumping the Kerberos Database" on page 231
- "Loading the Kerberos Database" on page 232
- "Stashing the Master Key" on page 233
- "Starting and Stopping Daemons" on page 235
- "Maintenance Tasks" on page 236
- "Backing Up primary security server Data" on page 237
- "Removing Unused Space from the Database" on page 239

Administering the Kerberos Database

After you have installed and configured the Kerberos server v3, the Kerberos database contains the default Kerberos principals, their keys, and other administrative information about each of these principals for your realm. For more information on installing your Kerberos server, see Chapter 2, “Installing the Kerberos Server v3.1,” on page 35.

You can use the Kerberos database utilities, `kadmin_ui` and `kadmin`, to globally manage the principals and their utilities. The `kadmin_ui` utility is the remote graphical user interface, and `kadmin` is the command-line interface.

The HP Kerberos Administrator is the graphical user interface that you can use to manage your principals and realms. This includes both the remote administrator, `kadmin_ui`, and the local administrator, `kadminl_ui`. This chapter contains a detailed description of both the utilities.

The remote command-line administrator, `kadmin`, contacts the Kerberos database, `kadmind`, for Kerberos authentication whereas the local administrator, `kadminl`, does not require a server for authentication. `kadminl` runs only on the primary security server where the Kerberos database is located. The Kerberos daemon, `kadmind`, is required for Kerberos authentication.

The kadmind Command

The `kadmind` command starts the administrative server. This administrative server runs on the Kerberos server that stores the Kerberos principal database. The `kadmind` command accepts password change requests and remote requests to administer the information in the Kerberos principal database.

Table 8-1 describes the configuration files that you must set for `kadmind` to work properly.

Table 8-1 Configuration Files Required for kadmind

File Name	Description
`/opt/krb5/krb.conf`	This is the Kerberos configuration file that contains configuration information for the Kerberos server. For more information, see “The krb.conf File” on page 65.
`/opt/krb5/krb.realms`	This is the Kerberos realms file that maps host names to their realm names. For more information, see “The krb.realms File” on page 66.
`/opt/krb5/admin_acl_file`	This is the access control list (ACL) of `kadmind` that lists the various principals along with their respective permissions. For more information, see “The admin_acl_file File” on page 113.
`/opt/krb5/password.policy`	This is the file that controls the password policy for all the principals. For more information, see “Password Policy File” on page 119.

The admin_acl_file File

The /opt/krb5/admin_acl_file file located only on the primary security server, lists authorized principals with their respective administrative permissions. It also lists principals that you cannot modify without explicit privileges.

NOTE Protect admin_acl_file with appropriate read-write privileges with access only to the root user

The kadmind command checks the permissions of the principal in admin_acl_file. You can edit admin_acl_file directly on the primary security server, or remotely using the Administrative Permissions window of the HP Kerberos Administrator.

The general format of admin_acl_file is as follows:

```
identifier/instance@REALM    [perms_list]      [# comments]
```

where:

identifier	Specifies the name of the principal.
instance	Specifies the administrative instance associated with the principal. HP recommends that you add an admin instance to each administrative principal name. If the prinicpal resides in the default realm of the primary security server, @REALM is optional. Otherwise, you must explicitly specify the realm of the principal.
[perms_list]	Specifies the permissions. You can add one or more permissions listed in Table 8-2, without any space between the letters.
[# comment]	Specifies any optional remarks about the principal. Characters after the # (hash) symbol are ignored.

Each line in admin_acl_file matches an administrative principal with a set of permissions. You can also use wildcards to enter groups of principal names.

Assigning Administrative Permissions

Administrative principals may have varying levels of trust assigned to them, depending on the policies of your organization.

Table 8-2 lists the possible administrative permission settings and the letter designator used in `admin_acl_file` to indicate the permissions assigned to the principal account.

Table 8-2 Administrative Permission Settings

Administrator Field Name	ACL File Character
Add principals.	`a` or `A`
Change principal passwords.	`c` or `C`
Delete principals.	`d` or `D`
Edit `admin_acl_file`. **NOTE:** You cannot restrict this setting by using the `r` or `R` permission.	`e` or `E`
Edit group defaults.	`g` or `G`
Inquire about principals. You can assign this attribute to all administrative principals to allow use of the administrative tools.	`i` or `I`
List principal. This is redundant with `i` or `I`. **NOTE:** This permission is not displayed in the HP Kerberos Administrator.	`l` or `L`
Modify principals.	`m` or `M`
Extract keys.	`x` or `X`
Restricted administrator. Use the `r`, `R`, and `Rr` modifiers in combination with `a, A, c, C, d, D, i, I, m, M,` or `x`. The `X` modifier allows you to permit administrative principals to use those options only against certain principals.	`r` or `R`

Permissions designated with a lowercase letter apply only to those realms to which the administrative principal belongs. Permissions designated with an uppercase letter apply to all realms. [permissions] is an optional string containing one or more options listed in Table 8-2.

The restricted administrator setting is a modifier that you must use in conjunction with permissions. You must consider the following guidelines before using the r, R and Rr modifiers:

- The order of the permission letters is irrelevant.
- The e, E, g and G switches are not affected by the r and R permissions.
- The * (asterisk) symbol overrides the r and R switches

For more information, see "Using Restricted Administrator" on page 117.

The principal can also include the asterisk (*) wildcard because admin_acl_file supports the following identifier/instance wildcards:

- */instance
- identifier/*

This format makes it easier to add groups of principal names to the file. Therefore, if you want any principal with the instance admin to have permissions to administer the database, you can use the principal */admin@REALM, where REALM is your realm of the primary security server.

For example, to grant all principals with the admin instance that need to have all the permissions assigned to them, add the following entry to admin_acl_file:

```
*/admin@FINANCE.BAMBI.COM *
```

where:

*	Denotes all prinicpals
admin	Specifies instance
FINANCE.BAMBI.COM	Denotes the realm name
*	Denotes permissions

To grant the principal rabbit@FINANCE.BAMBI.COM the permission to add, list, and inquire about any principal in the database, add the following entry to admin_acl_file:

```
rabbit@FINANCE.BAMBI.COM ali
```

Adding Entries to admin_acl_file

You can add any principal name to admin_acl_file with or without administrative permissions.

To add a principal with assigned permissions, select the **Principal Information window>Attribute tab** in the HP Kerberos Administrator. For more information, see “Administrative Permissions” on page 189.

Consider the following guidelines before deciding on the principal names that you want to add to admin_acl_file:

- A primary security server must contain only one admin_acl_file. This file contains all the realms supported by the primary security server.
- Any principal name that you add to admin_acl_file must have adequate protection because only trusted administrative principals must be able to alter the principal account using the remote administration tool.
- Principals in admin_acl_file that have assigned permissions can log on to the administrative tools and become administrative principals.

The r, R, or Rr modifiers, when used with the a or A permission, restrict the principal names that you can add to the database. For instance, principals assigned the IARiar permissions cannot add new principals that use the identifier/instance@REALM, which is already included in admin_acl_file. To take advantage of this restriction, consider the names you may want to add to admin_acl_file.

Creating Administrative Accounts

You can set administrative permissions in admin_acl_file using one of the following methods:

- Using the HP Kerberos Administrator to set administrative permissions. When you change the administrative permissions of the principal, admin_acl_file is automatically updated.
- Editing admin_acl_file directly. To edit this file, you need to have the required system file administration rights.

Using Restricted Administrator

The r, R, and Rr modifiers are used with the a, A, c, C, d, D, i, I, m, M, x, or X permissions to permit administrative principals to use those options only against certain principals.

How the r/R Modifiers Work

Consider the following factors while using the r, R, and Rr modifiers:

- The r modifier restricts only lowercase permissions. For instance, administrative principals with ird permissions cannot delete principals from their own realm that are included in admin_acl_file.

NOTE

The r modifier does not restrict upper-case permissions. For instance, administrative principals assigned with IMimr permissions cannot modify principals in their own realm that are included in admin_acl_file, but they are able to modify any principal in all other realms supported by the primary security server.

- The R modifier restricts only uppercase letter permissions and only applies to realms other than the realm of the administrative principal. For instance, administrative principals assigned the IRD permissions cannot delete principals included in admin_acl_file from any realm except their own.

NOTE

IRDid is equivalent to the IRD permissions because the uppercase permissions (excluding the r and R modifiers) apply to all realms.

In either case, administrative principals can delete any principal from their own realm, but they have restricted delete privileges in realms other than their own.

For example, administrative principals with IDRm or IDRidm permissions assigned have restricted delete permissions in all other realms but their own, but they can modify and delete any principal in their own realm.

- The Rr modifiers restrict permissions for all principals in admin_acl_file for all realms supported by the primary security server. For example, administrative principals with IMRimr permission assigned cannot modify principals included in admin_acl_file in any realm, including their own. They can only modify principals that are not included in admin_acl_file.
- The e, E, g, and G permissions are not affected by the r, R, and Rr modifiers.
- Administrative principals assigned with the icr or ICRicr permission are still able to change their own passwords using the administrative tools.

  Permissions other than c and C are restricted for the restricted administrative principals. For instance, principals assigned with the imr permission are not able to modify their own principal accounts.

  An administrative principal with r or R in combination with e or E can use the Kerberos administrative utilities to remove the r modifier from their admin_acl_file entry, for example: ier, IER, IERr, or IEr. Do not assign these permission combinations.
- Administrative principals assigned with the ic, icr, IC, or ICR permission are able to change principal attributes and extract service keys in addition to changing principal passwords. According to the r and R modifier rules, restricted administrators can only make the principal accounts, which are not included in admin_acl_file, perform these actions.

Password Policy File

The password policy file controls password rules, such as password length, number of character types, and the lifetime of a password. The `password.policy` file located on each of the primary and secondary security servers in the `/opt/krb5` directory.

Editing the Default File

To edit the password policy file and configure it to match the requirements of your organization, use a text editor on the primary security server. You must have the appropriate read-write permissions to access the password policy file.

The default password policy file is designed around the following instances or policy groups:

- Principals that do not have an instance
- Principals with an `admin` instance
- Principals with a root instance
- The base group named `*`, which consists of all the other principals

You can also add more policy groups to identify specific instances in your enterprise.

Table 8-3 explains the password policy settings and the defaults for the base group and the * instance group in the password policy file.

Table 8-3 Default Password Policy Settings for the Base Group

Password Policy Setting	Default Value
`*.MaxRepeatChars`	3
*.MaxRepeatClasses	4
`*.MaximumMatch`	4
`*.MinimumLength`	6
`*.MinimumClasses`	2

Table 8-3 Default Password Policy Settings for the Base Group

Password Policy Setting	Default Value
`*.Expiration`	None
`*.MinimumAge`	None
`*.NotifyTime`	7d
*.Dictionaries	None
`*.MaxFailAuthCnt`	10
*.NoReqChangePwd	0
`*.MaximumHistory`	1

If you modify the `MaxfailAuthCnt` parameter, you must copy the password policy file to the secondary security server and restart `kdcd` on both the secondary and primary secondary security servers.

NOTE `MaxFailAuthCnt` is the only parameter that the secondary security servers read in the password policy file.

If you edit the password policy file on the primary security server, the file must be copied to each secondary security server, so that all the servers have an updated version of this file.

For more information on the password policy file, type `man 4 password.policy` at the HP-UX prompt.

Principals

A principal is a specific entity to which you can assign a set of credentials. Principals are users and network services that are included in your security network.

The general syntax for a principal is as follows:

```
identifier/instance@REALM
```

where:

`identifier` Specifies the name of the network service or a user. This parameter is mandatory and you must specify the `identifier`.

`/instance` Specifies the group used to further identify the name. The `instance` can identify the duties, organization, or any other information about the principal.

For a user, the instance is often used to describe the intended use of the corresponding credentials.

For a host, the instance is the fully qualified domain name. You can specify up to 255 instances. You must precede each additional instance with a slash (`/`).

The commands `rlogind`, `ftpd`, `rshd`, `rcpd`, and `telnetd` use the instance to indicate the name of the system on which the network service resides.

An instance may also imply special privileges. For example, a security administrator can have a principal account with an `admin` instance to use when performing administration tasks.

The `/instance` parameter is not mandatory.

`Realm` Specifies the realm in which the principal resides. By convention, realm names are the fully qualified domain name of the primary security server.

This parameter is mandatory and you must specify the realm name.

When creating principal names, ensure that a principal name:

- Is case sensitive.
- Cannot be longer than 767 characters.
- Must be uniquely defined in the first 255 characters.
- Cannot contain a space, tab, pound symbol (`#`), backward slash (`\`) or colon (`:`).
- Does not subscribe to a `NULL` policy. If you subscribe to a policy that does not exist in the `password.policy` file, the default policy `*` is applied for the principal.

NOTE You can use the slash (`/`) character in a principal name to delineate an instance.

Following are the different types of principals:

- User principal

  A user principal is an account assigned to an individual in your organization. Each individual must have at least one account. You may choose to add multiple accounts for one individual if you intend to use the accounts for different purposes. Use the instance parameter of the principal name to designate the intended use of the account. Following are the special categories for user principals:

  Administrative principals are user accounts with administrative permissions assigned in `admin_acl_file`. HP recommends that you use the `/admin` instance to distinguish these accounts.

- Service principal

  A service principal is a principal account assigned to a service in your security network. Examples of service principals include secured daemons or services that are accessible on the network, and `host/` principals created for a host system of the user.

Adding User Principals

The Kerberos server enables you to add user principals to the principal database. The only limit on the number of principals in the database is the disk space available on the primary security server and on each of the secondary security servers.

When adding a user principal to the database, assign the principal identifier, instances (if used), and the realm name. You must also designate a temporary password for the principal. You may assign specific attributes and properties to the account. Any attributes and properties that are not specifically set for the principal are inherited from the default group principal.

Establish a secure method for transferring the temporary password information to the user to avoid a security breach. Communicate the temporary password before the user authenticates with the new principal account. Make sure the user knows that he or she is required to change the password during the first authentication attempt.

Adding New Service Principals

The Kerberos server enables you to add service principals to the principal database. Use service principal accounts for a UNIX host system, a Kerberos-secured service, or an application that is available to user principals in the network.

When the Kerberos server software is installed, the Kerberos server requires certain service principals that are automatically added to the principal database. You must manually add the service principal accounts used by the optional secured service applications to the principal database.

Each Kerberos-secured service or application must have the ability to provide its secret key during authentication. Therefore, service principal accounts must have the following specific attributes and properties, depending on the requirements of the application:

- The application must be able to provide its unique principal name during authentication.

The instance portion of the service principal name must be the fully qualified domain name (FQDN) of the host on which the service resides. Although the FQDN in your network can use mixed-case characters, the instance portion of the principal name must be in lowercase.

For example, if the system name is `IT.BAMBI.COM`, the principal name must use the instance `it.bambi.com`.

If you fail to use this principal naming convention for the Kerberos server utilities, daemons, and services, the service principals cannot authenticate, and other principals cannot access when required.

- You must set the `Allow as Service` attribute for the service principal account.
- You must extract the secret key to the service key table file on the host of the service. Unlike user principals who type their password using the keyboard, a service principal must have its secret key automatically available during authentication. Storing the key in the service key table file ensures that the key is available when required. For more information on extracting a key, see “Extracting Service Keys” on page 178.

Reserved Service Principals

The Kerberos server requires that certain service principals be included in the principal database. These principal accounts use reserved names that have a special significance in the Kerberos server database.

Most of these reserved service principals are automatically created when you create the principal database or add a realm to the database.

IMPORTANT Do not modify the password policy name of the reserved service principals.

This section contains a detailed description of the reserved service principals.

K/M@REALM: The `K/M@REALM` principal contains the secret key of the principal database. When creating the database, the Kerberos server adds the `K/M@REALM` principal to the default realm of the server to store

the database secret key. All records in the principal database are encrypted using this key. The key for this principal is stored on each Kerberos server in the `.k5.realm` file.

IMPORTANT Do not remove, modify, or change the key type for this principal. Do not generate a new key for this principal.

default@REALM: The `default@REALM` principal name contains the default group principal attributes for the realm. This principal is required in each realm. This principal, called the default group, is automatically created when a realm is added to the database.

The attributes and properties of this principal act as a template for adding principals to a realm in the principal database of the Kerberos server. This principal uses a random key. However, you must not extract this key to a service key table file. This principal is locked by default, eliminating the security risk of an external attack to authenticate using this principal account.

IMPORTANT Do not remove this principal entry or unlock this principal account.

krbtgt/REALM@REALM: You can use the secret key of the `krbtgt/REALM@REALM` principal to encrypt and decrypt ticket-granting tickets (TGTs) issued by the Kerberos server for principals in the `REALM`.

IMPORTANT Do not remove or modify this principal entry, except when adding a 3DES key if you need to add support for this encryption type.

To configure interrealm authentication, create distinct reserved principals with the prefix name `krbtgt/` for each realm.

If you change any attribute or password of the `krbtgt/REALM@REALM` principal for the default realm, that is, the realm that contains the `K/M@REALM` principal, you must close all administrative programs, including `kadmin`, `kadminl_ui`, and `kdcd`. Then, restart all administrative services and daemons in that realm for the changes to take effect.

kadmin/REALM@REALM: The Kerberos administrative graphical user interface and command-line interface utilities use the `kadmin/REALM@REALM` principal name. This principal is required in each realm. It automatically adds the principal name when you add a realm to the database.

This principal uses a random key, but you do not need to extract the key to a service key table file.

IMPORTANT Do not remove or modify this principal entry.

kadmin/changepw@REALM: The Kerberos v5 standard set/change password protocol requires the `kadmin/changepw@REALM` principal. This principal is automatically added to the database when a realm is created.

This principal uses a random key, but you do not need to extract the key to the service key table file.

IMPORTANT Do not remove or modify this principal entry.

kcpwd/REALM@REALM: The `kcpwd/REALM@REALM` principal name is the change password service for Kerberos. This principal is required in each realm. It is automatically added when you add a realm to the database.

This principal uses a random key. However, you do not need to extract this key to a service key table file.

IMPORTANT Do not remove or modify this principal entry.

host/fqdn@REALM: Kerberos servers and application services such as the following use the `host/fqdn@REALM` principal name:

- Primary and secondary security servers, depending on the requirement of the database propagation.
- Secure connection utility daemons and client applications.

You must enter the `fqdn` in lowercase letters, and the `fqdn` instance must be the fully qualified domain name of the host system for the server or service.

These principals are not automatically added to the principal database when you install the Kerberos servers or application services.

Removing User Principals

You may need to delete user principals from the database. When you delete a principal account from the database, the principal name, attributes, and properties are removed from the database and you cannot use the principal to authenticate to the Kerberos server. To delete a principal, use the HP Kerberos Administrator or the command-line interface administrative utility.

For user principals, you may need to perform additional steps to remove the special privilege settings.

For user principals that use a UNIX system, every UNIX host that a principal uses contains the `host/service` principal. If this system is unused, delete the service key from the host and remove the `host/<fqdn>` principal from the database.

Removing Special Privilege Settings

If the principal has special privileges, remove these privileges. Examples of special privileges are as follows:

- Administrative principal that is aware of the UNIX root password. Ensure that you change the root or administrator password according to your password requirements.
- Administrative principal using `kadmin`. Ensure that you remove the administrative principal entry in `admin_acl_file`.

NOTE

When you delete an administrative principal using the HP Kerberos Administrator, any reference to that principal is automatically removed from `admin_acl_file`.

Protecting a Secret Key

A user principal must provide its password during authentication to create the secret key of the user principal. For best security, all users must periodically change their passwords.

This version of Kerberos contains the following methods to enforce user principals to change their password:

- You can enable the `Password Change Required` attribute to enforce the users to change their passwords during next logon.
- When the password expiration date is exceeded, the user principal must change his or her password. The password policy file or the date set for the principal account using one of the Kerberos server administrative utilities contain the password expiration time.

In all these cases, users can use the UNIX command `kpasswd` to change their passwords. When users execute the `kpasswd` command at the HP-UX prompt, they must enter the current password, then enter the new password twice to verify the new password string. The new password of the principal is automatically checked against the password policy file to ensure that it meets the enterprise criteria for secure passwords. Using the password policy file, you can specify rules that require users to create passwords that can prevent easy discovery of the password. For more information on the Password Policy File, see “Password Policy File” on page 119.

If you are using a principal account with the required administrative permissions, you can change the password of the user principal without knowing the current password of the principal.

When you change the password of a principal using one of the Kerberos administrative utilities, the password is not verified against the password policy file. Therefore, after you set a password, the user must change the password the next time he or she attempts to authenticate using the account. The `Change Password Required` attribute is automatically enabled. You must securely communicate the temporary password to the user so that users are aware of their temporary passwords during next logon.

Removing Service Principals

When you delete a service principal account from the database, the service account is no longer available on the network.

Deleting a service principal using one of the Kerberos administrative utilities removes the principal name, attributes, and properties from the database.

For a service principal, you need to perform an additional step of removing its secret key, which is stored in the service key table file on the host of the service. This key is not deleted when the service principal is removed from the database. Therefore, you must manually delete the secret key from the database.

If a host contains only one service, you can delete the service key table file. The default name for the file is `v5srvtab`.

If multiple services share the same service key table file, remove the service key for the deleted service principal account from the service key table file. For information on deleting keys from the service key table file, see “Deleting Older Keys from the Service Key Table File” on page 245.

The kadmin and kadminl Utilities

The `kadmin` and `kadminl` Kerberos command-line administrative utilities provide a unified administration interface for the Kerberos database. Kerberos administrators use these utilities to create new users and services for the primary database, and to modify information for the existing entries in the database. You can use these utilities to maintain the Kerberos principals and service key tables (`v5srvtab`).

The `kadmin` utility is the remote Kerberos client. It resides on the secondary security servers and client systems. The `kadminl` utility is the local client; it resides on the primary security server and is intended for use by individuals with root access privileges. You can use `kadmin` to remotely maintain the database on the primary security server from the local workstation.

Alternatively, you can also use the GUI administrative utility `kadmin_ui` for remote administration and `kadminl_ui` for local administration.

Before you log on to the remote administrator from a secondary security server or use a client, add the administrative principal to the database on the primary security server.

To log on to the remote administrator, `kadmin`, use a principal account that has an entry in `admin_acl_file`. For complete access to all functions, use an unrestricted administrative principal account with the `*` permission in `admin_acl_file`. The account must have at least the `inquire` privileges. For more information on administrative permissions, see “The admin_acl_file File” on page 113.

For more information on the `kadmin` option, type `man 1 kadmin` at the HP-UX prompt.

Administration Utilities

Table 8-4 describes the administrative utilities that you can use to administer the Kerberos database.

Table 8-4 Administration Utilities

Name	Description
`kadminl_ui`	The local graphical interface that runs on the primary security server.
`kadminl`	The local command-line administrator that runs on the primary security server.
`kadmin_ui`	The remote GUI that can only be run by administrative principals with the required permissions. It runs on all secondary security servers and any client system where the utility is installed.
`kadmin`	The remote command-line administrator that can only be run by administrative principals with the required permissions. It runs on all secondary security servers and on any client system where the utility is installed.
`krb5_encrypt`	The `krb5_encrypt` tool encrypts the password with the master key that is located in the stash file.

NOTE You cannot use the command-line administrator to control administrative permissions, maximum ticket lifetimes and renew times or the addition of new realms. Therefore, HP recommends that you use the GUI administrative utility for all administrative purposes.

HP Kerberos Administrator

HP Kerberos Administrator is a graphical user interface that you can use to administer the principal database.

You can use the HP Kerberos Administrator to perform the following functions:

- Creating, modifying, and deleting principals.
- Altering a principal account key type setting.
- Assigning administrative permissions.
- Modifying the default group principals.
- Extracting principals to service key table files.

Following are the different types of graphical user interface administrative utilities:

- `kadminl_ui` –The local administrator.
- `kadmin_ui` – The remote administrator.

The local administrator, `kadminl_ui`, is available only on the primary security server, and is located in the `/opt/krb5/admin/kadminl_ui` directory.

You can install the remote administrator, `kadmin_ui`, on secondary security servers and clients to permit remote administration of the principal database. The `kadmin_ui` remote administrator is located in the `/opt/krb5/bin/kadmin_ui` directory on the secondary security servers and clients.

NOTE You must use `kadminl` to add the first administrative principal, before you can use `kadmin` from a secondary security server or client.

You need not log on as an `admin` principal to the local administrator. A user with `root` access to the primary security server can run `kadminl`.

Alternatively, to log on to the remote administrator, use a principal account that has an entry in `admin_acl_file`. For complete access to all the functions, use an unrestricted administrative principal account with

the * permissions in admin_acl_file. The account must have at least inquire privileges. For more information, see “The admin_acl_file File” on page 113.

Both the local and remote administrators are discussed in detail in this chapter.

Standard Functionality of the Administrator

This section contains the standard descriptions applicable to all windows in the HP Kerberos Administrator window.

Table 8-5 describes the function of the OK, Apply and Cancel buttons.

Table 8-5 Function of OK, Apply, and Cancel Buttons

Button Name	Action
OK	Click **OK** to save your changes and to close the active window. If the window has multiple tabs, the changes for all the tabs are saved.
Apply	Click **Apply** to saves your changes and to retain the active window open. If the window has multiple tabs, the changes for all the tabs are saved.
Cancel	Click **Cancel** to discard the changes and to close the active window. If the window has multiple tabs, then the changes that you have made in all the tabs are also discarded.

Local Administrator – kadminl_ui

The local administrator, `kadminl_ui`, is the GUI-based database administrator that runs on the primary security server. It allows principals with administrative privileges to administer and maintain the principal database on an ongoing basis. You must include all the users, clients, or services authenticated by the Kerberos server as principals in the principal database.

You can use `kadminl_ui` to perform the following functions:

- Adding, modifying, listing, or deleting principals.
- Searching for principals.
- Listing administrative permissions.
- Extracting service principal keys to the key table files.
- Adding, modifying, listing, or deleting realms.
- Editing the default group.

Using kadminl_ui

This section provides an abridged summary on using the `kadminl_ui`. Execute the following at the HP-UX command prompt to invoke the HP Kerberos Administrator window:

```
# /opt/krb5/admin/kadminl_ui
```

The HP Kerberos Administrator window contains the following tabs:

- Principals
  - — Managing Principals
  - — Using Groups to Control Settings
  - — Setting Administrative Principals
- Realms
  - — Adding a Realm
  - — Deleting a Realm

This chapter contains a detailed description of the Principals tab and the Realms tab.

Principals Tab

You can use the Principals tab (Figure 8-1) in the HP Kerberos Administrator window to manage principal entries in your database by adding, editing, or deleting principals.

Table 8-6 describes the components of the Principals tab.

Figure 8-1 Principals Tab

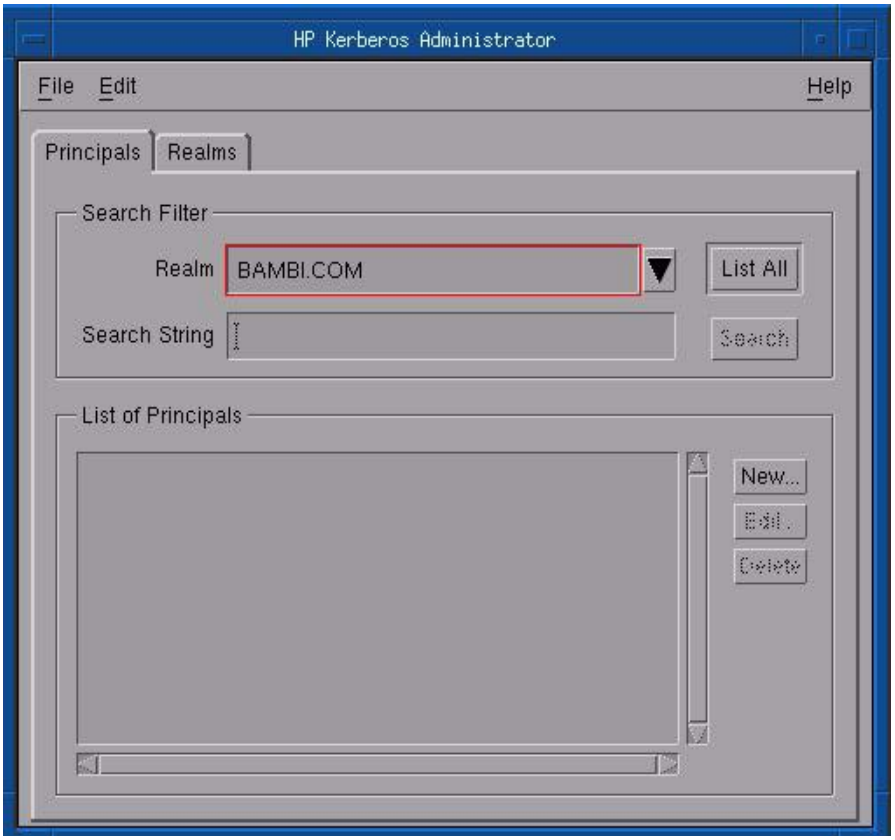


Table 8-6 Principals Tab Components

Component Name	Description
Realm	Select the realm where the principal that you want to add, change, or delete resides.

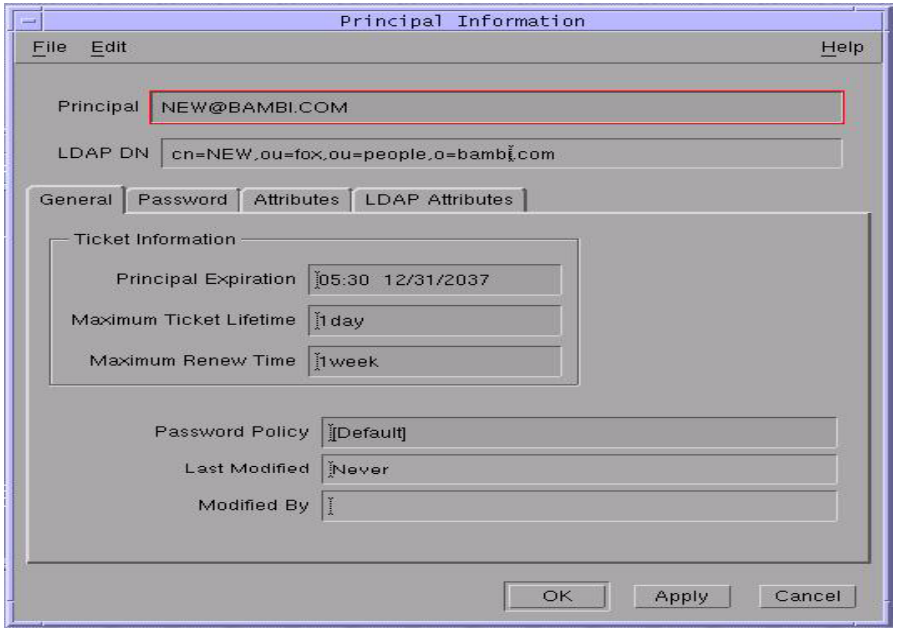
Table 8-6 Principals Tab Components (Continued)

Component Name	Description
List All	Click this button to list all the principals associated with the realm. **NOTE:** If you have selected LDAP as the backend database, then information about all realms under the same base DN is displayed when you click this button.
Search String	Enter characters for locating the principal that you want to edit or delete. See "Searching for a Principal" on page 149 for more information on searching a principal.
Search	Click this button to list all the principals matching the search criteria that you enter in Search String. See Table 4 on page 149 for more information.
List of Principals	Display a list of all the principals in the realm or those that match your search criteria.
New	Create a new principal.
Edit	Change an existing principal. You must select a principal in the List of Principals dialog box to enable this button.
Delete	Deletes an entry. Select an entry in the List of Principals dialog box to enable **Delete**.

General Tab (Principal Information Window)

You can use the Principal Information window to add principals or to modify existing principals and ticket information. To add a new principal, select the realm in the HP Kerberos Administrator window and click **New**. The Principal Information window displays as shown in Figure 8-2.

Figure 8-2 Principal Information Window



You can use the Principal Information window to specify the Principal name and the LDAP DN.

Table 8-7 describes the components of the Principal Information window.

Table 8-7 Principal Information Window Components

Field Name	Description
Principal	You can use this field to enter the name of the new principal.

Table 8-7 Principal Information Window Components (Continued)

Field Name	Description
LDAP DN	Displays the LDAP DN.
General Tab	You can use the General tab on the Principal Information window to specify the ticket information, the password policy file, and values for Last Modified and Modified By, for a new or existing principal.
Password Tab	You can use the Password tab on the Principal Information window to specify the password parameters for the principal. Table 8-10 describes the components of the Password tab.
Attributes Tab	You can use the Attributes tab in the Principal Information window to assign attributes for a principal. Table 8-12 describes the components of the Attributes tab.
LDAP Attributes Tab	You can use the LDAP Attributes tab in the Principal Information window to assign LDAP attributes for a principal. This tab is disabled when you are editing an existing Kerberos principal.

Table 8-8 describes the components of the General tab.

Table 8-8 General Tab Components

Field Name	Description
Principal	Displays the name of the principal that you are editing. If you are adding a new principal, enter the name of the new principal.

Table 8-8 General Tab Components (Continued)

Field Name	Description
Principal Expiration	Displays the principal expiration time, which indicates when the current logon privileges of the principal expire. Enter one of the following options in the Principal Expiration box: • A date and time in the format `HH:MM MM/DD/YYY`. • The keyword `Never`, which indicates that the principal does note expire.
Maximum Ticket Lifetime	Sets the maximum ticket lifetime for a principal. The time must not exceed the corresponding ticket lifetime set for the realm, as controlled by the `krbtgt/REALM@REALM` principal.
Maximum Renew Time	Sets the maximum renewal time for each individual principal. You must specify a time that does not exceed the corresponding renewal time set for the realm. To restrict the issuance of renewable tickets, uncheck the **Allow Renewable** option on the Principal Information window>Attributes tab. You must not enter a renewal time of 0 (zero) seconds. If you enter 0 seconds for the Maximum Renew Time field in the General tab and click **OK** or **Apply**, the following warning message is displayed: `Invalid Field Entry`

Table 8-8 General Tab Components (Continued)

Field Name	Description
Password Policy	Specifies the password policy name in this field. If you do not specify the password policy name, the default policy is applied. **NOTE:** Do not change the password policy name for reserved service principals.
Last Modified	Specifies the last modified date for the principal. The last modified date and time is automatically updated when you change the principal information.
Modified By	Specifies the name of the user updating the principal.

Adding Principals to the Database

When you add a principal, you must specify the following information:

- Principal and ticket information, located in the General tab.
- Password and password expiration information, located in the Password tab.
- Other principal attributes, located in the Attributes Tab

The Group Settings option controls the default values in all the tabs in the Principal Information window. Choose **Principal Information>Edit>Load Default values** to reload default at any time, while working on principal. For more information, see “Loading Default Values for a Principal” on page 152.

To add a new principal, complete the following steps:

Step 1. In the HP Kerberos Administrator window, select the **Realm** in which you want to add a new principal.

Step 2. Click **New** to display the Principal Information window as shown in Figure 8-2.

Step 3. In the General tab, enter the Principal name and the Ticket Information.

Step 4. Click **Apply** in the General Tab. The Change Password window displays as shown in Figure 8-3.

Figure 8-3 Change Password Window

Step 5. Enter the new password in the Change Password window and click **OK**.

Step 6. In the Password tab, enter the Password Information and the Key and Salt Types.

You cannot use the Change Password button in the Password tab because the Change Password button is disabled when you create a new principal. For more information, see “Changing Password Information” on page 158.

Step 7. In the Attributes tab, select the required principal attributes. For more information, see “Changing Principal Attributes” on page 167.

Step 8. Click **OK** in the Principal Information window.

The List of Principals in the HP Kerberos Administrator window is not automatically refreshed after you add a principal. The database contains the principal, but you need to click **List All** or **Search** to view the principal in List of Principals.

Adding Multiple Principals with Similar Settings

To simultaneously add multiple principals with the same setting, complete the following steps:

Step 1. In the HP Kerberos Administrator window, select the **Realm** in which you want to add multiple principals.

Step 2. Click **New** to display the Principal Information window as shown in Figure 8-2.

Step 3. Enter the first principal name in the Principal Information window.

Step 4. Select the targeted settings for the principal in the General, Password, and Attribute tabs of the Principal Information window.

Step 5. Click **Apply** to display the Change Password window.

Step 6. Enter the initial password for the new principal and click **OK**.

The new principal entry is saved in the principal database and the Principal Information window is still open

Step 7. Enter another principal name in the Principal Information window, and repeat the steps 5 through 7 until you finish adding all the prinicpals.

Step 8. When you have added all the new principals with the common attributes, click **OK** or **Cancel** to close the Principal Information window.

For additional information about adding a principal, see “Loading Default Values for a Principal” on page 152 and “Setting Administrative Permissions” on page 188.

Creating an Administrative Principal

You can use the HP Kerberos Administrator window to create an administrative principal. When you create a principal and assign the administrative permissions to it, the principal is stored in `admin_acl_file` located on the primary security server. For more information on `admin_acl_file`, see “The admin_acl_file File” on page 113.

HP recommends that you assign the `/admin` instance to a principal that is an administrator. Therefore, a user can have one or more of the following principal accounts:

- One or more principals with non administrative permissions for daily authentication.
- One principal account with the `/admin` instance that has administrative permissions

.

NOTE The /`admin` principal of the user must have a different password than the password for other principal accounts of the user. This provides additional security during administrative tasks.

To create an administrative principal, complete the following steps:

Step 1. In the HP Kerberos Administrator window, select the **Realm** in which you want to create an administrative account.

Step 2. Click **New** to display the Principal Information window as shown in Figure 8-2.

Step 3. Enter the `identifier/admin@REALM` of the administrative principal in the Principal field.

Step 4. In the General tab, the default ticket information for the administrative principal already exists. You may change this information or retain the old values.

Step 5. Click **Apply** to display the Change Password window as shown in Figure 8-3.

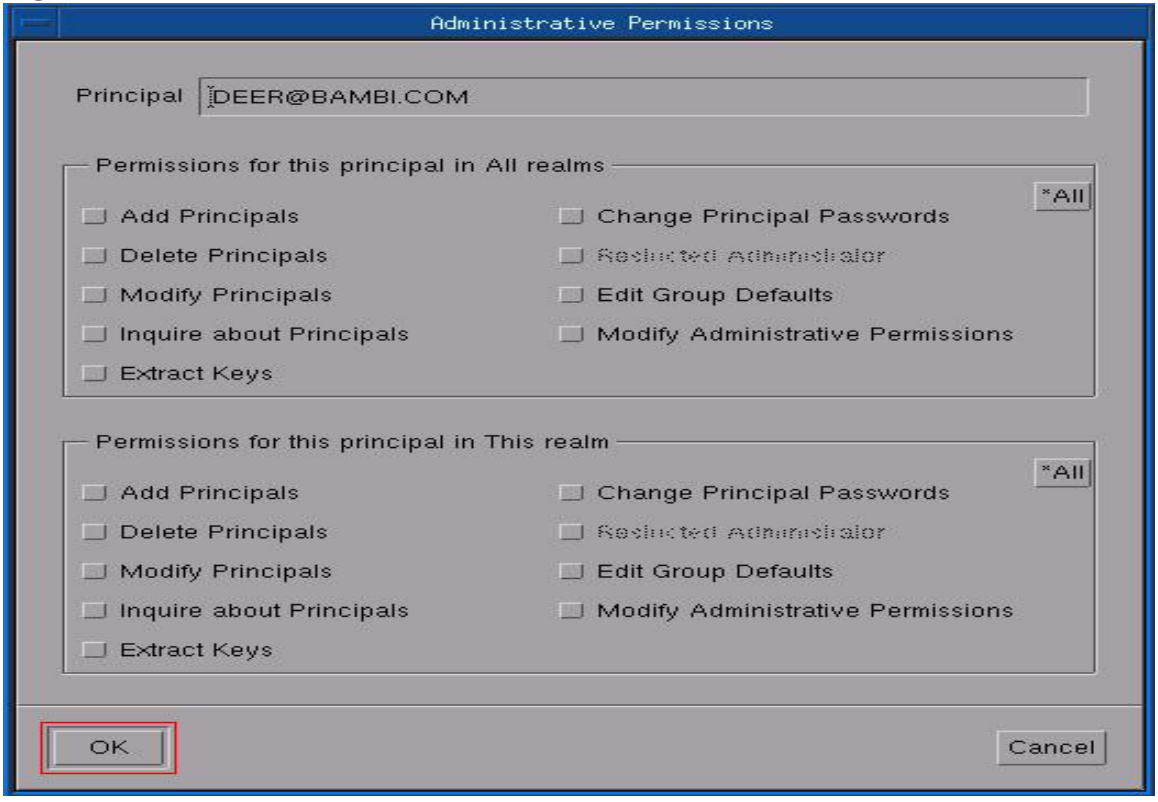
Step 6. Enter the password information and click **OK** in the Change Password window. Do not select the Generate Random Key option.

Step 7. In the Attributes tab, select the attributes for the administrative principal. Select the **Require Preauthentication** attribute if the administrative principal requires a hardware authentication device.

Step 8. If necessary, click **Apply**.

Step 9. Select **Principal Information>Edit>Edit Administrative Permissions** to display the Administrative Permissions window as shown in Figure 8-4.

Figure 8-4 Administrative Permissions Window



Select the permissions for the administrative principal.

Step 10. Click ***All** to select all the permissions and click **OK**.

Step 11. Click **OK** to save all the values to the database and to close the Principal Information window, or click **Cancel** to close the Principal Information window without saving the values to the database.

Searching for a Principal

Following are the methods to search for a principal:

- Click **List All** in the Principals tab to display a list of principals in the current realm in the List of Principals list box, which displays up to 1,000 principals.
- Click **Search** to display a list of principals in the current realm that match the search criteria in the Search String text box. The List of Principals dialog box displays up to 1,000 principals.

To search for a principal, complete the following steps:

Step 1. In the Principals tab, select the **Realm** in which you want to search a principal.

Step 2. Enter the search criteria for the principal in the Search String text box. The search string can contain wildcards.

Step 3. Click **Search** to display the principal names that match your search criteria in the current realm.

The List of Principals list box displays a list of 1,000 principal names in the current realm that match your search criteria.

Step 4. If you cannot find the principal name, refine your search using the search criteria described in Table 8-9.

Table 8-9 Search Criteria

Character	Description
*	Represents any string of characters including / (slash). For example: • */admin searches for all principal names with an admin instance. • *test* searches for all principals containing the characters test.
?	Each ? represents a single character except / (slash). For example: • test? searches for all principal names containing the characters test followed by a single character. • test?? searches for all principal names containing the characters test followed by any two characters.

Table 8-9 Search Criteria (Continued)

Character	Description
[...]	Represents any one character from the set except / (slash). For example, `[abc]*` searches for all principal names starting with `a`, `b`, or `c`. The following characters have a special meaning with the `[...]` construct: **!** Represents an exclusion when used immediately after `[`. For example: `[!abc]*` searches for all principal names that start with any character other than `a`, `b`, or `c`. `[]a]*` searches for all principal names that start with the `character ]` or `a`. `[!]a]*` searches for all principal names that do not start with the character `]` or `a`. **-** Represents a range of characters. When used immediately after `[` or `[!`, or as the last character in the set, this character is matched literally. For example: `test [1-9]` searches for all principal names starting with the characters `test`, followed by an one number is the range 1 through 9. `[abc-]` searches for all principal names starting with `a`, `b`, `c`, or, "-". **:** Represents the starting of a character class descriptor when used immediately after `[` or `[!`. Otherwise, it is matched literally. The following character classes are supported: `[:alnum:] [:alpha:] [:cntrl:] [:digit:] [:graph:] [:lower:] [:print:] [:punct:] [:space:] [:upper:] [:xdigit:]`

Deleting a Principal

When you delete a principal using one of the Kerberos administrative utilities, all references to the principal are automatically removed from both the principal database and `admin_acl_file`.

To delete a user principal, complete the following steps:

Step 1. In the Principals tab, select the **Realm** in which you want to delete a principal.

Step 2. Click **List All** or enter the **Search String** to find the principal.

Step 3. Select the principal name from the List of Principals list box, and click Delete. The following warning message displays:

```
Do you really want to delete principal <principal_name>?
```

Step 4. Click **OK** to delete the principal.

The List of Principals list box is automatically updated after the principal is deleted.

Loading Default Values for a Principal

When you add or edit a principal in the Principal Information window, you can quickly restore any changed values to the default values that are specified in the default group. When you reload the default values, all fields in the Principal Information window are updated with the default values.

To view or modify the current default settings, select **Edit>Edit Default Group** in the Principal Information window.

To reload the default values for a principal, complete the following steps:

Step 1. In the Principals tab, select the **Realm** in which you want to update the principal.

Step 2. Select the principal in the List of Principals list box for which you want to load the default value.

Step 3. Click **Edit** to display the Principle Information window as shown in Figure 8-2.

Step 4. Choose **Edit >Load Default Values** in the Principal Information window.

The values in all the Principal Information window tabs are reset to the default values.

Restoring Previously Saved Values for a Principal

You can restore any value for a principal that you have changed but not yet saved to the values that were previously saved for that principal.

To retain the previously saved values for a principal while you are modifying the values for that principal in the Principal Information window, choose **Edit>Restore Values** in the Principal Information window. The values in all the tabs of the Principal Information window are restored to the previously saved values for this principal.

Changing Ticket Information

You can change the ticket information used for a principal, including the principal expiration date, ticket lifetime, and ticket renewal time.

To change the ticket information, complete the following steps:

Step 1. In the Principals tab, select the **Realm** where the principal resides.

Step 2. Click **List All** or **Search** to find the principal whose ticket information you want to change.

See “Searching for a Principal” on page 149 for more information on how to search for a principal.

Step 3. Click **Edit** to display the Principal Information window as shown in Figure 8-2.

Step 4. In the General tab, change the principal expiration date using the required format.

For more information on the date format and changing ticket information, see “Rules for Setting Maximum Ticket Lifetime” on page 155 and “Rules for Setting Maximum Renew Time” on page 156.

NOTE You only need to edit the ticket lifetime for a principal if that principal must have a maximum ticket lifetime shorter than the `krbtgt/REALM@REALM` principal.

Step 5. Click **OK** to accept the new ticket information.

You can use the Principal Information>Edit options to quickly reload default values or to restore previously saved values at any time while editing a principal. When you reload default values or restore previously saved values, all the fields in the Principal Information window tabs are reset to the default or previously saved values.

Rules for Setting Maximum Ticket Lifetime

Maximum ticket lifetime indicates the maximum lifetime for which a ticket can be issued to the principal. You can specify the maximum ticket lifetime value in the **General>Maximum Ticket Lifetime** text box.

The format for the ticket lifetime is as follows:

```
[Nw] [Nd] [Nh] [Nm]
```

where:

N	Indicates an integer.
`w, d, h, m`	Identifies the unit of time: weeks, days, hours, or minutes, respectively.

A number without a suffix `w,d,h`, or `m` is interpreted as hours.

NOTE Spaces are not allowed between the number and the suffix.

Following are some examples for denoting the maximum ticket lifetime:

- `1d3h` is 1 day and 3 hours.
- `9h18m` is 9 hours and 18 minutes.
- `26` is 26 hours.

You can also express the time units by using full words. For example, `1day` is the same as `1d`. You can also use the keywords `week`, `day`, `hour`, and `minute` to denote `w`, `d`, `h`, or `m`, respectively.

The maximum lifetime for a ticket issued to any principal in any given realm is controlled by the settings for the reserved principal `krbtgt/REALM@REALM`.

Rules for Setting Maximum Renew Time

Maximum renew time indicates the maximum amount of time for which a ticket can be renewed. You can specify the maximum renew time value in the **Principal Information>General>Maximum Renew Time** text box.

The format for the ticket lifetime is as follows:

```
[Nw] [Nd] [Nh] [Nm]
```

where:

N	Indicates an integer number.
`w, d, h, m`	Identifies the unit of time: weeks, days, hours, or minutes, respectively.

A number without a suffix `w,d,h`, or `m` is interpreted as hours.

NOTE Spaces are not allowed between the number and the suffix.

Following are some examples for denoting the maximum renew time:

- `1d9h` is 1 day and 9 hours.
- `4h24m` is 4 hours and 24 minutes.
- `18` is 18 hours.

You can also express the time units by using full words. For example, `1day` is the same as `1d`. You can also use the keywords `week`, `day`, `hour`, and `minute` to denote `w`, `d`, `h`, or `m`, respectively.

The maximum lifetime for a ticket issued to any principal in any given realm is controlled by the settings for the reserved principal `krbtgt/REALM@REALM`.

Do not enter a renewal time of 0 (zero). To restrict the issuance of renewable tickets, clear the **Allow Renewable** option in the Principal Information **window>Attributes** tab.

The following error message is displayed when you enter 0 (zero) for Maximum Renew Time and click **OK** or **Apply** in the Principal **Information>General** tab:

```
You have entered an invalid time
```

Changing Password Information

You can change the following password information used by a principal:

- Password expiration date

  Indicates when the password of the current principal is due to expire. Check the **Password Expiration Date** box to activate password expiration for the current principal. If you do not enable this option, the current password of the principal will never expire.

- Key version number

  Indicates the number of times a password has been changed by the principal.

- Key types

  Indicates the key encryption options, namely, `3DES`, `DES-MD5` and `DES-CRC`.

- Salt types

  Indicates the value used by the secret algorithm to create a unique encrypted key even for identical passwords within a single system. Salts also allow unique keys on two separate systems to use the same password.

To change the password information, complete the following steps:

Step 1. In the HP Kerberos Administrator window, choose the **Principals** tab.

Step 2. In the Principals tab, select the realm where the principal resides.

Step 3. Click **List All** or **Search** to find the principal.

Step 4. Select the principal name from List of Principals and click **Edit** to display the Principal Information window as shown in Figure 8-2.

Step 5. In the Principal Information window>Password tab, change the password information for the current principal.

Step 6. Click **OK** to accept the new password information.

IMPORTANT If you change the key or salt type, you must change the password of the principal. You must inform the principal of the required temporary password. The principal must change the password during next logon.

You can use the `Principal Information>Edit` menu to quickly reload default values or to restore previously saved values anytime while editing a principal. When you reload default or restored values, all the fields in the Principal Information window tabs are reset to the default or previously saved values, respectively.

Password Tab (Principal Information Window)

You can use the Password tab (Figure 8-5) on the Principal Information window to specify the password parameters for the principal.

Figure 8-5 Password Tab

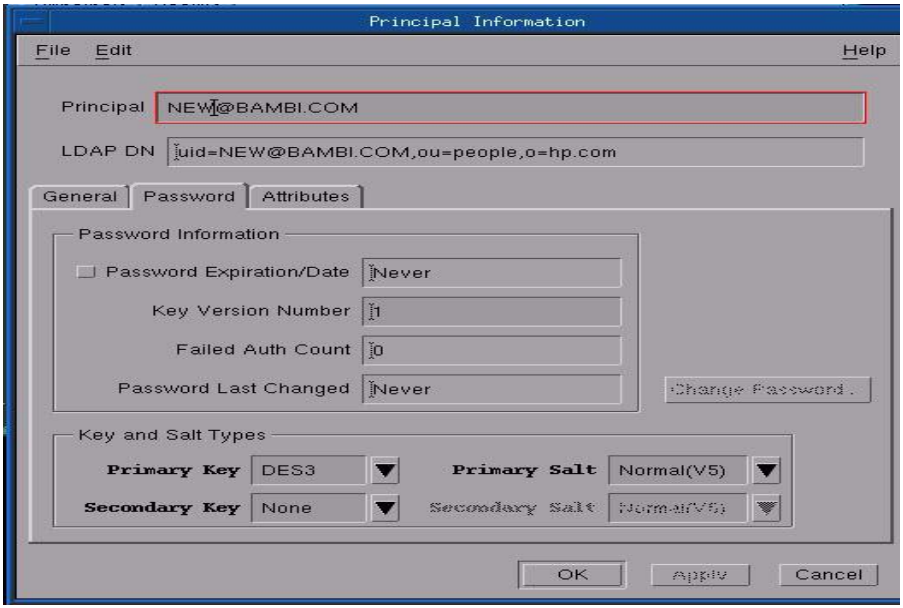


Table 8-10 describes the components of the Password tab.

Table 8-10 Password Tab Components

Component Name	Description
Principal	Displays the name of the principal that you are editing.
LDAP DN	Displays the LDAP DN that you are editing.

Table 8-10 Password Tab Components (Continued)

Component Name	Description
Password Expiration/Date	Indicates when the current principal password expires. Select **Password Expiration/Date** to activate password expiration for the current principal. If you do not enable this function, the password of the current principal never expires. You can enter one of the following options in the Password Expiration/Date text box: • A date and time, in the following format: `HH:MM MM/DD/YYY` • The keyword `NEVER`, which indicates that the principal will never expire.
Key Version Number	Specifies the version number of the principal password. The version number identifies the number of times the password has been changed. When a principal is created for the first time, its password version number is 1. Every time the password is changed, the version number is incremented by 1.
Password Last Changed	Indicates the date when the password was last changed. The keyword `NEVER` indicates that a user cannot change the password.
Change Password	Displays the Change Password window, which provides the option to specify a password or to generate a random key for the current principal.

Table 8-10 Password Tab Components (Continued)

Component Name	Description
Failed Auth Count	Specifies the number of failed authentication attempts since the last successful authentication by the principal. Every failed `SignOn` request by the client increments the Failed Auth Count value by 1. If the number exceeds the maximum value allowed by the `MaxFailAuthCnt` parameter in the password policy file, the principal account is automatically locked. To determine if a principal account is locked, click **Principal Information>Attributes** and verify if `Lock Principal` check box is selected. To unlock a principal, clear the check box.
Primary and Secondary Key Types	Specifies the available key encryption options: `DES3`, `DES-MD5`,and `DES-CRC`. Select a key encryption type for each salt type that you use.
Primary and Secondary Salt Types	Specifies the salt type for a principal. A `Salt` is a string of characters added to a password before it is transformed into the secret key. Each salt type, except `None`, has some data associated with it. The salt data is appended to the password before generating the `DES3` or `DES` encrypted key. The salt key settings are controlled through the Password tab. Salts are used to strengthen passwords and to ensure that principals with the same passwords do not have the same key.

Change Password Window (Password Tab)

When you create a new principal using the `Principal Information window>Password` tab, HP Kerberos Administrator automatically displays the Change Password window (Figure 8-6). Enter a new password and verify the password for user principals. You must select

Generate Random Key only for service principals. If you select the Generate Random Key option, a unique encrypted key is created without entering a password.

Figure 8-6 Change Password Window

Table 8-11 describes the components of the Change Password window.

Table 8-11 Change Password Window Components

Components	Description
Principal	Displays the name of the principal that you are editing.
Generate Random Key	Generates a random key for a principal. Generating a random key always results in a unique key. Random keys are generally created for service principals.

Table 8-11 Change Password Window Components

Components	Description
New Password	Specifies the new password information. This is a temporary password because the principal is required to change the password of the user during next logon. The assumption is that the `NoChangeReqPwd` setting in the password policy file of the principal is set to 0 (zero), which is the default.
Verification	Re-enter your new password for verification.

Changing a Key Type

For a strong enterprise wide security between the Kerberos servers and clients, all principals must have 3DES keys using Normal (V5) salt.

Changing a DES-CRC or DES-MD5 Principal Key Type to 3DES

If you are changing the key type for a service principal that has extracted keys, complete the following steps on the host system where the service resides:

Step 1. Log on using a principal account that contains the required administrative permission, and launch the remote administrator, HP Kerberos Administrator.

Step 2. In the HP Kerberos Administrator window, choose the **Principals** tab and select the realm of the principal.

Step 3. Click **List All** or **Search** to find the principal.

Step 4. Select the principal name from List of Principals and click **Edit** to display the Principal Information window as shown in Figure 8-2.

Step 5. Choose the **Password** tab in the Principal Information window.

Step 6. Under the Key and Salt Types, select the primary and secondary key types and salt types. If the principal was formerly **DES-CRC** or **DES-MD5** principal, you can retain one key as **DES** and set the other key to **3DES**.

Step 7. Click **OK**.

The Change Password window appears because you must generate a new password if you change the key or salt type.

NOTE

Consider the following points while changing the password:

- If the principal is a user principal, enter a new password.

- If the principal is a service principal with an extracted key, select the **Generate Random Key** check box to generate a random key.

Step 8. Click **OK** to close the Change Password window.

Step 9. Click **OK** to close the Principal Information window.

If the principal is a user principal, communicate the new temporary password to the user. During next logon, the principal must change the password.

If the principal is a service principal, extract the new key for the principal. For more information on this procedure, see “Extracting Service Keys” on page 178.

Changing Principal Attributes

You can change the attributes of a principal in the Principal Information window (Figure 8-5). These attributes are the characteristics and properties assigned to a user or a service principal. Attributes control how a principal behaves and what that principal can do, such as renewing and forwarding valid tickets.

To change the principal attributes, complete the following steps:

Step 1. In the HP Kerberos Administrator window, choose the **Principals** tab and select the realm of the principal.

Step 2. Click **List All** or **Search** to find the principal.

Step 3. Select the principal name from List of Principals, and click **Edit** to display the Principal Information window as shown in Figure 8-2.

Step 4. Choose the **Attributes** tab to change the attributes for the current principal. For more information on attributes, see “Attributes Tab (Principal Information Window)” on page 168 and “Password Tab (Principal Information Window)” on page 160.

Step 5. Click **OK** to accept the new attributes and to close the Principal Information window.

You can use the **Principal Information window>Edit>Load Default Values** option to quickly reload default values or restore previously saved values at any time while editing a principal. When you reload default or restored values, all the fields in the Principal Information window tabs are reset to the default or previously saved values.

Attributes Tab (Principal Information Window)

Attributes are the characteristics and properties assigned to a principal that control the behavior of the principal. You can use the Attributes tab in the Principal Information window to assign attributes for a principal, as shown in Figure 8-7.

Figure 8-7 Attributes Tab

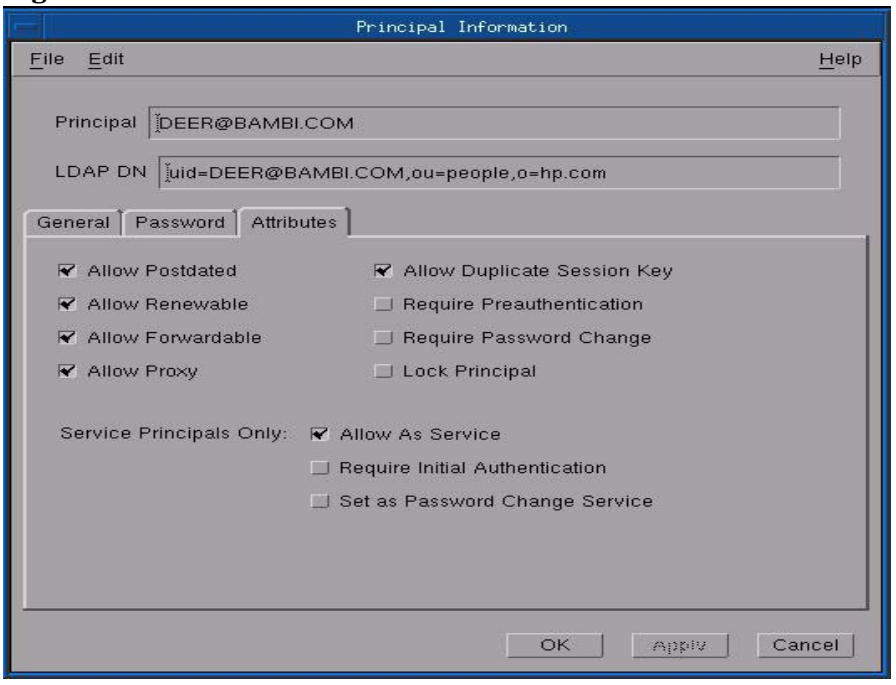


Table 8-12 describes the components of the Attributes tab.

Table 8-12 Attributes Tab Components

Components	Description
Principal	Displays the name of the principal you are editing.

Table 8-12 Attributes Tab Components (Continued)

Components	Description
LDAP DN	Displays the LDAP DN that you are editing.
Allow Postdated	Specifies whether a principal is allowed for ticket postdating. Postdating is a mechanism that allows a principal to obtain a ticket that is initially invalid, but that can become valid at some time in the future. The Allow Postdated attribute applies to both user and service principals. If you set the attribute for a user principal, the user can be issued a postdated or postdatable ticket. If you set the attribute for a service principal, the server can issue postdated service tickets for the service.
Allow Renewable Tickets	Specifies if a principal is allowed to renew tickets. Renewable tickets are those that a principal can revalidate up to the maximum renewable time. The Allow Renewable attribute applies to both user and service principals. If you set the attributes for a user principal, the principal can be issued a renewable ticket. If you set this attribute for a service principal, the server issues a renewable ticket for the service. You can set the maximum renewable time in the Principal Information>General tab.

Table 8-12 Attributes Tab Components (Continued)

Components	Description
LDAP DN	Displays the LDAP DN that you are editing.
Allow Postdated	Specifies whether a principal is allowed for ticket postdating. Postdating is a mechanism that allows a principal to obtain a ticket that is initially invalid, but that can become valid at some time in the future. The Allow Postdated attribute applies to both user and service principals. If you set the attribute for a user principal, the user can be issued a postdated or postdatable ticket. If you set the attribute for a service principal, the server can issue postdated service tickets for the service.
Allow Renewable Tickets	Specifies if a principal is allowed to renew tickets. Renewable tickets are those that a principal can revalidate up to the maximum renewable time. The Allow Renewable attribute applies to both user and service principals. If you set the attributes for a user principal, the principal can be issued a renewable ticket. If you set this attribute for a service principal, the server issues a renewable ticket for the service. You can set the maximum renewable time in the Principal Information>General tab.

Table 8-12 Attributes Tab Components (Continued)

Components	Description
Allow Forwardable	Specifies if a principal is allowed ticket forwarding. Forwarding is a process that sends a ticket-granting ticket (TGT) from one network host to another host. The second host system can use the forwarded TGT to generate a new service ticket on behalf of the principal. The Allow Forwardable attribute applies to both user and service principals. If you set this attribute for a user principal, the principal can be issued a forwarded or forwardable ticket. If you set this attribute for a service principal, the server can issue a forwarded service ticket for the service.
Allow Proxy	Specifies if a principal is allowed proxy tickets. Proxy tickets allow applications that a principal accesses with a TGT to request a special class of service ticket. You can move this type of service ticket to another host on the network that acts on behalf of the principal, for example, a print service printing a file. The Allow Proxy attribute applies to both user and service principals. If you set this attribute for a user principal, the principal can be issued a proxy ticket. If you set this attribute for a service principal, the server can issue a proxy service ticket for the service.
Allow Duplicate Session Keys	Specifies if a principal is allowed to use a duplicate session key. A duplicate session key is used in user-to-user authentication and specifies which key is used to encrypt the tickets.

Table 8-12 Attributes Tab Components (Continued)

Components	Description
Require Preauthentication	Specifies if a principal is required to use preauthentication in the TGT request. Preauthentication means that additional known encrypted data is sent with the ticket request, providing additional security when the TGT is presented to gain access to a secured service. The Require Preauthentication attribute applies to user and service principals. If this attribute is set for a user principal, the user is required run the logon software that performs authentication using the preauthentication protocol. If this attribute is set for a service principal, the service cannot accept TGTs from a user principal if the user did not obtain a TGT using a preauthentication protocol.
Require Password Change	Specifies that a principal must change its password during the next logon to the Kerberos server. The Require Password Change attribute applies to user principals. When new principals are added to the database or when the password of the principal is changed, this attribute is controlled by the `NoReqChangePwd` setting in the password policy file of the principal. By default, `NoReqChangePwd` is set to 0 (zero), meaning that users must change their passwords during first logon.

Table 8-12 Attributes Tab Components (Continued)

Components	Description
Lock Principal	Specifies if a principal is active. A locked principal still exists in the principal database, but it is unable to use or provide Kerberos services. The Lock Principal attribute applies to both user and service principals. If you set this attribute for a user principal, tickets cannot be issued to the user. If you set this attribute for a service principal, tickets are not issued to it. When a principal exceeds the maximum number of failed authentication attempts allowed by the password policy file, the Lock attribute is set. The default maximum level allowed for failed authentication attempts is 5. If a principal is locked, an administrative user must unlock the principal before the user authenticates.
Allow As Service	Specifies if a principal is allowed to act as a service. Set this attribute to allow a principal to act as a service (that is, the name of the principal is in the server field of the service ticket). You must select this attribute for any principal that is used as a service principal. You can apply the Allow As Service attribute to all principals, in addition to principals that act solely as service principals. The attribute is selected by default. **NOTE:** User principals must have this attribute set when using user-to-user authentication.

Table 8-12 Attributes Tab Components (Continued)

Components	Description
Require Initial Authentication	Specifies if the server is allowed to issue service to the service principal on behalf of a user principal using a previously obtained TGT. If you set this attribute for the service principal, a user principal must authenticate to the server again, to obtain a ticket for that service. For example, the Change Password service requires that a principal enter a password to receive a ticket for the Change Password service. If you do not set this attribute, the server issues a server ticket based on the TGT that it already possesses. The Require Initial Authentication attribute applies only to service principals. If you select this attribute for a principal being edited or created, the Allow as Service attribute is automatically selected.
Set As Password Change Service	Specifies if the server is allowed to issue initial tickets to this service principal for user principals whose passwords have expired.

LDAP Attributes Tab (Prinicpal Information Window)

The LDAP Attributes tab displays the mandatory LDAP attributes that need to be specified while creating a Kerberos principal. These attributes need to be specified only if the LDAP DN does not exist in the Directory server. The mandatory attributes displayed in this tab are based on the default object class template that you specified while configuring your Kerberos server with LDAP as the backend.

The LDAP attributes tab is not displayed if the default object class template consists of only one mandatory attribute. Also, if you have entered an LDAP DN that already exists in the Directory server, then the LDAP attributes tab is disabled.

NOTE You can view only nine mandatory attributes of the default object class template in the LDAP attribute tab. By default, the remaining mandatory attributes are set to their default values.

You can use the LDAP Attributes tab in the Principal Information window to assign LDAP attributes for a principal, as shown in Figure 8-8.

Figure 8-8 LDAP Attributes Tab

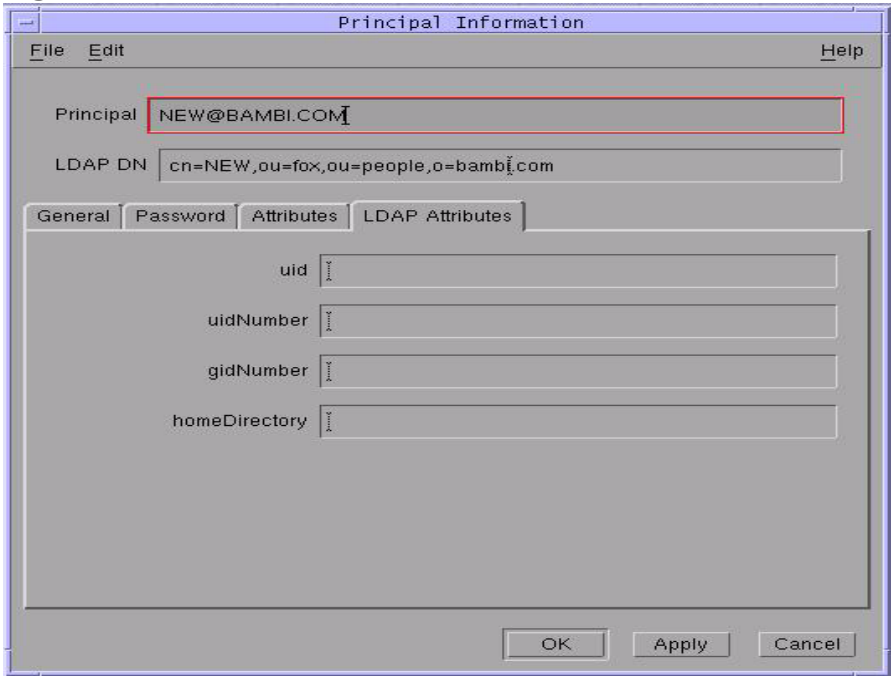


Figure 8-8 describes the components of the LDAP Attributes tab, if you have opted for the object class `posixaccount`. The LDAP Attributes tab displays the mandatory attributes based on the object class selected.

NOTE You must enter values for all the mandatory attributes in the LDAP Attributes tab; else a warning message is displayed.

Deleting a Service Principal

The Kerberos server requires several specific principals. If you accidentally delete these principals, you must restore the principal database from a backup tape.

To delete a service principal that has a random key extracted to the service key table, remove the entry from the `v5srvtab` service key table file by completing the following steps:

Step 1. Log on to the host system where the service is located, or connect to the remote host using the `telnet <host_name>` command.

Step 2. Launch the remote administrator, HP Kerberos Administrator, and log on using a principal account that has the required administrative permissions.

Step 3. In the HP Kerberos Administrator window, choose the **Principals** tab.

Step 4. In the Realm text box, select the realm that contains the service principal to be deleted.

Step 5. Select the principal name to be deleted in the List of Principals list box.

Step 6. Click **Delete** to delete the principal.

The following warning message is displayed:

`Do you really want to delete principal <principal_name>?`

Step 7. Click **Yes** to delete the principal.

The principal is automatically deleted from the List of Principals list box.

To remove the entry of the principal from the key table, `v5srvtab`, use the `ktutil` tool. For more information on `ktutil`, type `man 1 ktutil` at the HP-UX prompt.

Extracting Service Keys

Unlike users who type their password using a keyboard, a service principal needs to have its secret key automatically available during authentication. Therefore, store the secret key for the service principals on the host where the service is located, in the service key table called the `v5srvtab` file.

The service key table, `v5srvtab`, contains service principal names and their corresponding keys. Typically, secret keys are randomly generated for the service key table file on the host system where the service resides. Therefore, the key can be obtained from the service key table when the service is invoked.

You must have administrative permissions to add and delete principals to extract the principal key to the service key table.

To extract principal keys securely to the service key table, complete the following steps:

Step 1. Log on to the host system where the service is located, or connect to the remote host using the `telnet` <*host_name*> command.

Step 2. Launch the remote administrator, HP Kerberos Administrator, and log on using a principal account that has the required administrative permissions.

Step 3. In the HP Kerberos Administrator window, choose the **Principals** tab and select the realm of the principal.

Step 4. Click **List All** or **Search** to find the principal.

Step 5. Select the principal name from **List of Principals** and click **Edit**. The Principal Information window displays as shown in Figure 8-2.

Step 6. Select **Principal Information>Edit>Extract Service Key** to display the Extract Service Key to Service Key Table window. (Figure 8-9).

Step 7. In the Extract Service Key to Service Key Table window, type the path and file name for the service key file in the Service Key Table Information box.

If you change the default name and location to a different name and location than the programs of the Kerberos server, you must edit the settings to indicate the new location of the service key table file.

Step 8. Select the **Generate New Random Key before Extracting** option. HP recommends that you select this option for increased security because it generates a new random key before the principal and key are extracted to the service key table.

Step 9. Click **OK** to extract the principal and its key to the service key table. If a service key table file does not exist in the selected directory, a new file is created. You cannot create a service key if the selected directory does not exist.

Consider the following points while extracting principal keys to the service key table:

- HP recommends that you re-extract all the service keys once a month, thereby changing the keys and reducing the risk of compromise to the keys.
- If the host system contains more than one service principal account, extract the service key for each principal individually.
- The extracted key is appended to an existing service key table file. If the extracted key has the same principal name as an existing table entry, the old key is overwritten with the new extracted key.
- Extracting a random key may modify the salt types of the principal whose key is being extracted. This is a normal side effect of generating a random key because a random key implies a salt type of v5 (none).

Extracting a Service Key Table

You can extract the key for a service principal to the service key table (`v5srvtab`) by using the Extract Principal Key to Service Key Table window. Because a service does not enter the password using the keyboard, you must store its secret key in a table. When this service is invoked, the key is obtained from the table.

Choose **Principal Information>Edit** and select the **Extract to Service Key Table** option to display the Extract Service Key Table window. (Figure 8-9).

Figure 8-9 Extract Service Key Table Window

Table 8-13 describes the components in the Extract Service Key Table window.

Table 8-13 Extract Service Key Table Components

Component	Description
Principal	Displays the name of the principal for which you are extracting a key.
Service Key Table Type	Identifies the type of key table into which the principal name and keys are extracted.
Service Key Table Name	Displays the default location and name of the service key table. Type a new service key table name in this box to use a service key table other than the default `v5srvtab` file. If you change the default name, also edit the `KeyTab` environment variable setting on the host of the service.
File	Selects a new path or name for the service key table file.
Generate New Random Key before Extracting	Generates a new random service key for the principal before extracting it to a service key table. HP recommends that you use this option for high security.

Using Groups to Control Settings

You can modify the default values used for new principals using the Principal Information window (Figure 8-2). Each realm has a default group, and the default group for the realm contains default values.

The values that you specify for the default group are inherited by all new principals in that realm. A default group is similar to a template that controls the settings for a new principal. This template is called `default@REALM`. When a new realm is created, this template is automatically added to the principal database.

While adding or editing a single principal, you can elect to override the default values of the group. If the default values set during installation are not what you commonly use for new principals, edit the default group for the realm. When you edit the template settings, settings for the previously created principals are not changed.

Editing the Default Group

To edit the default group, complete the following steps:

Step 1. In the HP Kerberos Administrator window, choose the **Principals** tab and select the realm of the principal.

Step 2. Click **New** to open the Principal Information window as shown in Figure 8-2.

Step 3. Choose **Principal Information>Edit**, and select the **Edit Defaults Group** option to display the Group Information window (Figure 8-10).

Step 4. In the Group Information window, set the new default values in the General tab, the Password tab, and the Attributes tab. The fields in these tabs are identical to the fields in the corresponding Principal Information window tabs (Figure 8-2).

Step 5. Click **OK** to close the Group Information window.

Step 6. Click **OK** to close the Principal Information window.

You can also edit the default group by selecting the `default@REALM` principal from the List of Principals list box in the Principals tab. In the Principals tab, click **Edit** to open the Principal Information window, and enter the value for all the fields in the General, Password, and Attributes tabs.

Group Information Window (Principal Information Window)

You can view or modify the default group settings of a realm using the Group Information window. The default group is similar to a template used to control the settings for new principals added to a realm. Changing the settings for the default group does not affect the settings for existing prinicpals in the realm.

The tabs and fields in this window correspond to the tabs and fields in the Principal Information window. The values that you enter in the Group Information window are provided as default values for new principals added in a realm. You can edit the settings for existing principals by using the tabs in Principal Information window.

To open the Group Information window, choose **Principal Information>Edit>Edit Default Group** to display the Group Information window (Figure 8-10).

Figure 8-10 Group Information Window

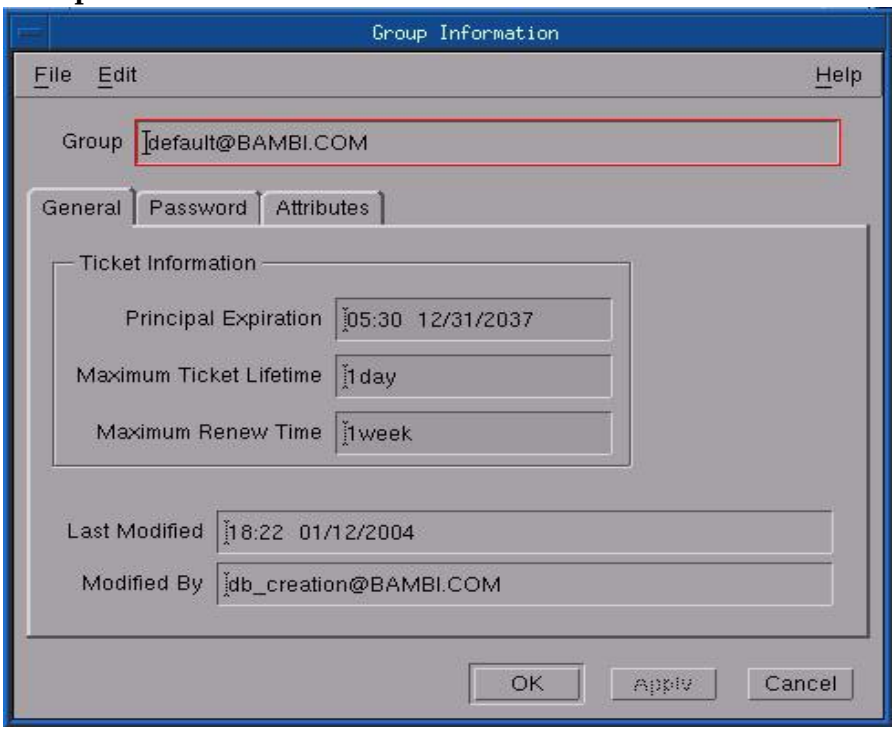


Table 8-14 describes the components of the Group Information window.

Table 8-14 Group Information Window Components

Component	Description
Group	Displays the default group principal name.

Table 8-14 Group Information Window Components

Component	Description
Group Information window tabs	Displays the following tabs: • "General Tab (Principal Information Window)" on page 139 • "Password Tab (Principal Information Window)" on page 160 • "Attributes Tab (Principal Information Window)" on page 168 The fields in these tabs are similar to the tabs in the Principal Information window (Figure 8-2).

Principal Attributes

You must assign attributes to each principal to control the usage and rights of the account. This section describes the possible attributes and the default settings.

Setting the Default Group Principal Attributes

Before adding principals to the database, you must review the attributes and properties assigned to the default group principal, `default@REALM`, and establish a set of base attributes for the principal. This default group principal acts as a template, and when the new principals are created, they inherit the attributes of the default principal.

NOTE Modifying the default group principal does not update the attributes of existing principals. Therefore, you must consider the assigned attributes before you begin adding principals.

Default Principal Attributes

When you create a realm in a database, the `default@REALM` principal is automatically created. Every realm contains one default principal that controls the settings for the default group.

To edit the default group, use the HP Kerberos Administrator or the command-line administrator, discussed as follows:

- In the HP Kerberos Administrator window, complete the following steps to edit the default group:

  1. Select a principal from the realm.
  2. Click **Edit** to open the Principal Information window.
  3. Choose **Principal Information>Edit**, and select the **Edit Default Group** option to display the Group Information window (Figure 8-10).
  4. In the Group Information window, you can edit the settings for the default group.

- In the command-line administrator, `kadminl`, manually edit the `default@REALM` principal.

  You cannot edit the key and salt type setting for the default group. For more information on using the command-line administrator, type `man 1M kadmin` or see “Manual Administration Using kadmin” on page 202.

Setting Administrative Permissions

Use the HP Kerberos Administrator window to assign administrative permissions to users. When you assign administrative permissions to a principal, the principal and its permissions are saved to `admin_acl_file` located on the primary security server.

HP recommends that you add the `/admin` instance to a principal to identify a principal as an administrator. The `/admin` instance of the user must have a password different from the password for other instances, thus providing additional security during administrative tasks. To perform administrative tasks, you must log on to the HP Kerberos Administrator with the `admin` principal, for instance, `user/admin@REALM`.

To set administrative permission, complete the following steps:

Step 1. In the HP Kerberos Administrator window, choose the **Principals** tab and select the realm where the principal is located.

Step 2. Click **List All** or **Search** to find the principal for which you want to assign administrative permissions. For more information on how to search a principal, see “Searching for a Principal” on page 149.

Step 3. Click **Edit** to display the Principal Information window (Figure 8-2).

Step 4. Choose the **Principal Information>Edit**, and choose the **Edit Administrative Permissions** option to display the Administrative Permissions window (Figure 8-11).

Step 5. In the Administrative Permissions window, select the appropriate permissions for the principal. You can assign permission for the principal for all realms or only for the realm where the principal resides.

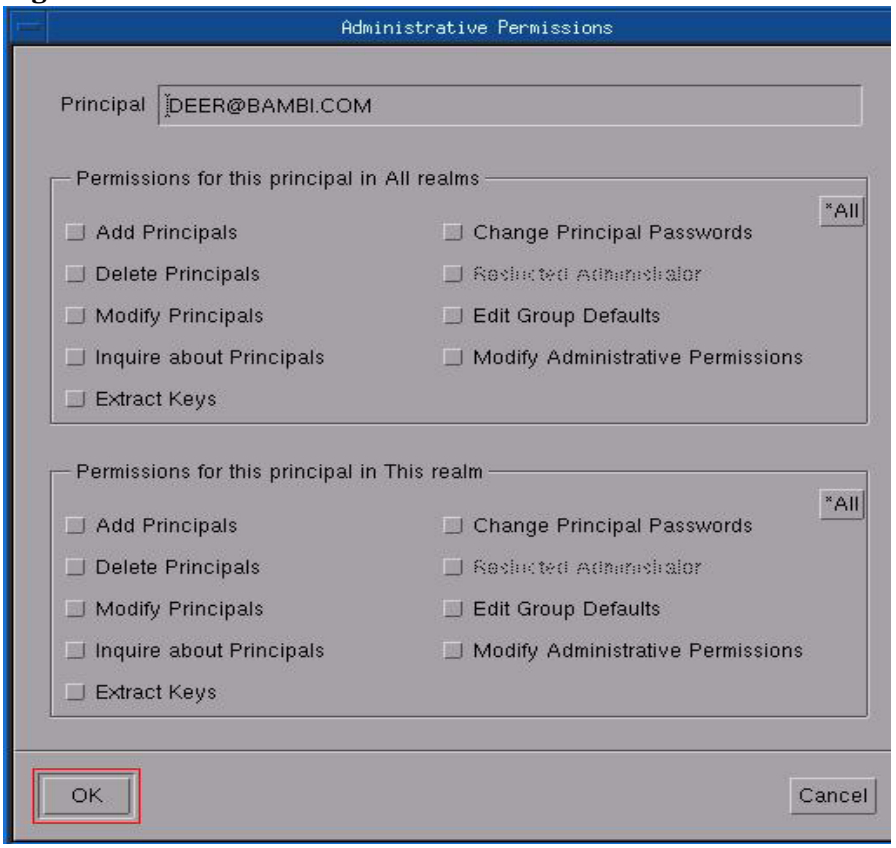
To enable a principal to run the remote or local administrative utility, you must enable the Inquire About Principals option. For more information, see “Administrative Permissions” on page 189.

Step 6. Click **OK** to save the permissions to `admin_acl_file`.

Administrative Permissions

You can assign administrative permissions using the Administrative Permissions window. Choose **Principal Information>Edit**, and select the **Edit Administrative Permissions** option to display the Administrative Permissions window (Figure 8-11). You can assign different levels of permissions in any combination or all the permissions to a single user.

Figure 8-11 Administrative Permissions Window



You can assign eight permissions in the KDC for the current realm or for all realms. If you assign permissions only for the current realm, the administrator can perform administrative tasks only within the realm. You can use the Restricted Administrator modifier in conjunction with

the Add Principals, Delete Principals, Change Principal Password, Inquire About Principals, Modify Principals, and Extract Keys permissions.

Table 8-15 describes the components of the Group Information window.

Table 8-15 Group Information Window Components

Component	Description
Principal	Displays the name of the principal that you are editing. You must add an additional principal account with the `/admin` instance for the individual requiring administrator privileges.
Add Principals	Allows the principal to add new principals to the principal database.
Delete Principals	Allows the principal to delete principals from the principal database.
Modify Principals	Modifies principals.
Inquire about Principals	Inquires about specific prinicpals. You must enable this option for all principals that access the Kerberos administrative utilities. If you wish to log on to the HP Kerberos Administrator, select the **Permission for this principal in This Realm box>Inquire about Principals** attribute.
Extract Keys	Extracts a key into the service key table file.
Change Principal Password	Changes principal passwords. In addition to the principals in `admin_acl_file`, you can also change any principal in the principal database.

Table 8-15 Group Information Window Components (Continued)

Component	Description
Restricted Administrator	Select this option in addition to the Add Principals, Delete Principals, Modify Principals, Inquire about Principals, Extract Keys, Change Principal Password attributes in the realm of the administrative principal or all realms to permit administrative principals to use these options only for the following principals: • Restricted administrator in the This Realm box – Restricts actions on `admin_acl_file` entries that belong to the realm of the administrative principal. • Restricted administrator in the All Realms box – Restricts actions on `admin_acl_file` entries that belong to realms other than the realm of the administrative principal. • Restricted administrator in both the This Realm box and the All Realms box – Restricts actions on `admin_acl_file` entries that belong to any realm supported by the primary security server. You cannot restrict the administrative principals that have the Restricted Administrator modifier from managing principals that are not included in `admin_acl_file`. The Restricted Administrator modifier setting does not override the Modify Administrative Permissions, that is, an administrative principal with both the Modify Administrative Permissions and the Restricted Administrator settings can change the principal settings in `admin_acl_file`, including their own principal settings. The Restricted Administrator modifier setting also does not override the **Principal Information>Edit>Edit Group Default** setting; an administrative principal with both these settings enabled can edit the values of the default group principal.
Edit Group Defaults	Edits the default values stored in the default group for the realm. You can edit the default principal using the **Principal Information>Edit>Edit Default Group>Group Information** window.

Table 8-15 Group Information Window Components (Continued)

Component	Description
Modify Administrative Permissions	Modifies administrative permissions for others users. You can modify the administrative permission using the **Principal Information>Edit>Edit Administrative Permissions>Administrative Permissions** window.
All*	The Administrative Permissions window contains two All* buttons for the following purposes: • A designated principal in all realms • A designated principal in the specified realm Click **All*** to assign all administrative permissions for the principal in all realms or only in this realm.

Realms Tab

A realm is a collection of principals that reside in the same administrative domain. Your network-naming scheme, network topology, security policy, and company organization determine which principals and services you put in a relam. Within a realm, all principals share the same security and administrative policy.

You can open the HP Kerberos Administrator window by executing the following command at the HP-UX prompt:

```
#  /opt/krb5/admin/kadminl_ui
```

Click the **Realms** tab to display it as shown in Figure 8-12. You can use the Realms tab to add or delete realms.

Figure 8-12 Realms Tab

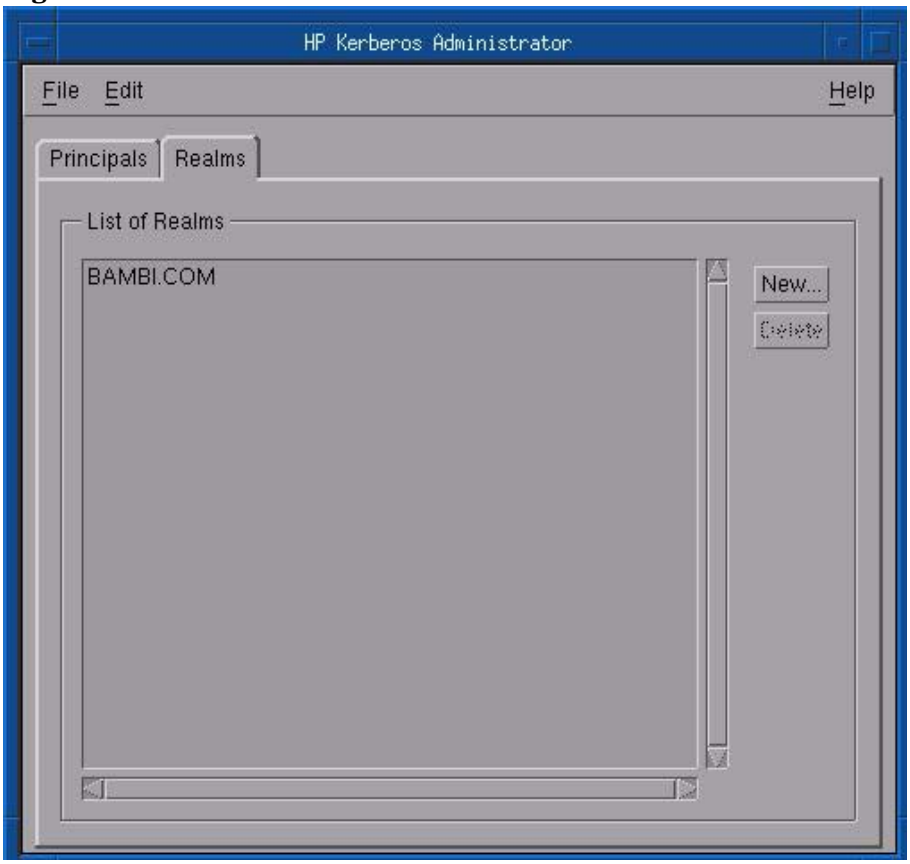


Table 8-16 describes the components in the Realms tab.

Table 8-16 Realms Tab Components

Component	Description
List of Realms	Displays a list of all the available realms.
New	Creates a new realm.
Delete	Deletes a realm. You must select an entry to enable this button.

Realm Information Window

You can use the Realm Information window to add realms. Click **New** in the HP Kerberos Administrator window>Realms tab to display the Realm Information window as shown in Figure 8-13.

Figure 8-13 Realm Information Window

Table 8-17 describes the components of the Realm Information window.

Table 8-17 Realm Information Window Components

Component	Description
Realm	Name of the new realm. Before you enter a new realm name, consult the realm architecture created during installation for the realm-naming scheme adopted by your organization.

Adding a Realm

When you add a realm, HP Kerberos Administrator automatically creates some reserved principals, which remain in the database.

To add a realm, complete the following steps:

Step 1. In the HP Kerberos Adminsitrator window, select the **Realms** tab (Figure 8-12).

Step 2. In the Realms tab, click **New** to display the Realm Information window (Figure 8-13).

Step 3. Enter the name of the new realm in the Realm field.

Step 4. Click **OK** to create the new realm.

If you want to include the realm in the interrealm authentication, create the ticket-granting service principal for interrealm authentication. See Chapter 10, “Managing Multiple Realms,” on page 275, for more information on interrealm authentication.

Deleting a Realm

When you delete a realm, all the principals for that realm are not deleted from the database. To delete the principals from the database, you can use the HP Kerberos Administrator window or the command-line administrator. For more information, see “Deleting a Principal” on page 151.

To delete a realm, complete the following steps:

Step 1. In the HP Kerberos Administrator window, select the **Realms** tab (Figure 8-12).

Step 2. In the List of Realms box, select the realm to be deleted.

Step 3. Click Delete to delete the realm. The following warning message displays requesting a confirmation for the delete operation:

```
Do you really want to delete realm <realm_name>?
```

Step 4. Click **OK** to delete the realm.

Remote Administrator – kadmin_ui

The `kadmin_ui` utility is the GUI-based Kerberos remote administrative utility that runs on the secondary security servers and clients.

Principals with administrative privileges use the remote administrator to maintain the database on the primary security server from their local workstation.

You can use `kadmin_ui` to accomplish the following tasks:

- Add, modify, specify, or delete principals.
- Search for a principal.
- List administration permissions.
- Extract service principal information to the service key table.
- Add, modify, specify, or delete realms.
- Edit the Default Group.
- Manage `admin_acl_file`.

You must have a principal with administrative privileges to use the `kadmin_ui` utility. The functionality of the `kadmin_ui` is identical to that of `kadminl_ui`. See “Local Administrator – kadminl_ui” on page 134 for more information on how to administer the Kerberos database locally.

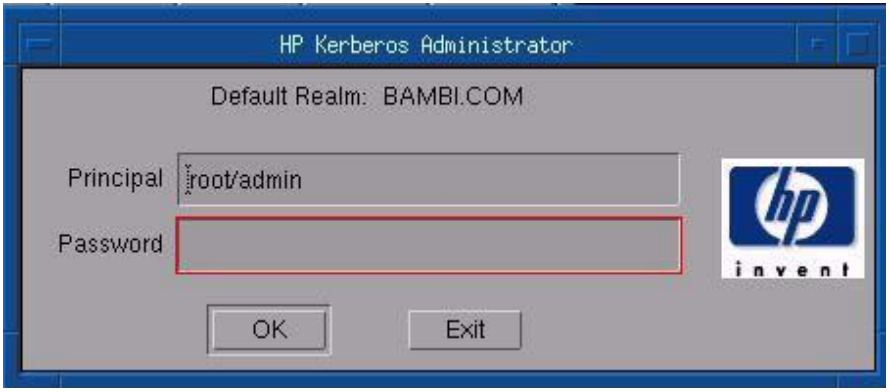
The following steps describe how to remotely administer the Kerberos database using the `kadmin_ui` window:

Step 1. Execute the following command at the HP-UX prompt:

```
# /opt/krb5/kadmin_ui
```

The logon screen displays as shown in Figure 8-14.

Figure 8-14 Logon Screen

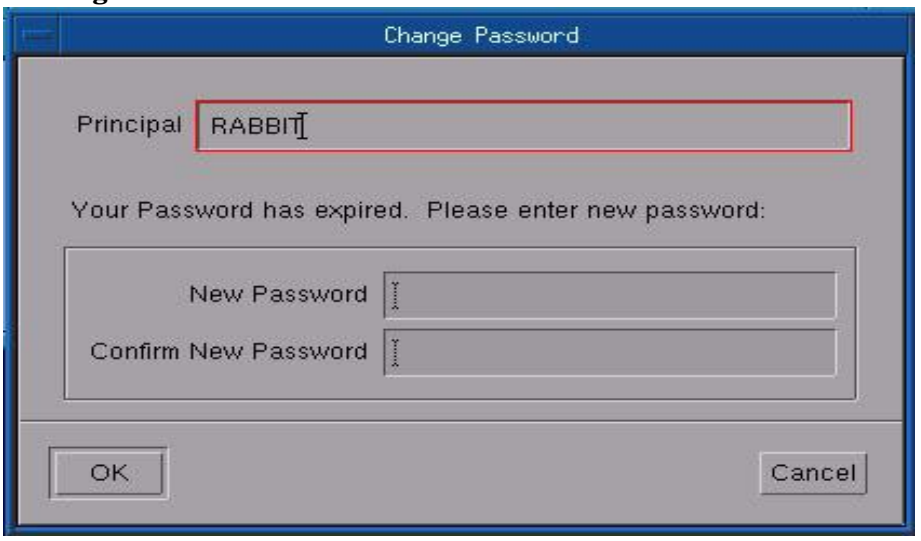


Step 2. Enter your principal name and password in the logon screen.

Step 3. Click **OK** to display the change password window as shown in Figure 8-15.

Step 4. Enter a new password in the change password screen to change your password, and click **OK**.

Figure 8-15 Change Password Screen



NOTE This screen is displayed only when you first log on using the remote administrator.

To access the database using the remote administrator, you must have the appropriate administrative permissions. The warming message, shown in Figure 8-16, displays if you do not have administrative privileges.

Figure 8-16 Warning Message

The graphical user interface for the remote administrator is similar to that for the local administrator. For more information on using the remote administrator, `kadmin_ui`, and administering your Kerberos server, see “Local Administrator – kadminl_ui” on page 134.

Manual Administration Using kadmin

You can use the command-line administrator to administer the principal database. It enables principals with administrative privileges to maintain the principal database. You must include all the users, clients, and services authenticated by the Kerberos server into the principal database.

The following types of command-line interfaces are available:

- Local command-line administrator, `kadminl`
- Remote command-line administrator, `kadmin`

The local command-line administrator, `kadminl`, is available only on the primary security server. You can install the remote command-line administrator, `kadmin`, on the secondary security servers and clients to remotely administer the principal database.

The local administrator, `kadminl`, is situated in the following directory on the primary security server:

```
/opt/krb5/admin
```

The remote administrator, `kadmin`, is situated in the following directory on secondary security servers and clients:

```
/opt/krb5/bin
```

NOTE You must add the first administrative principal on the local administrator, `kadminl`, located on the primary security server before you log on to the remote command-line administrator, `kadmin`, from a secondary security server or client.

You can use `kadmin` to perform the following tasks:

- Add, modify, inquire, or delete principals.
- Change the password of an existing principal.
- Extract a key for an existing prinicpal.
- Extract service principal information to the service key table.

Only a user with `root` permission can invoke the local command-line administrator, `kadminl`.

To log on to the remote administrator, `kadmin`, use a principal account that has an entry in `admin_acl_file` and an account that has at least `inquire` privileges. For complete access to all functions, use an unrestricted administrative principal account, one with the `*` permissions in `admin_acl_file`. The account must at least have `inquire` privileges. For more information on administrative permissions, see "The admin_acl_file File" on page 113.

When you start `kadmin`, you must specify a principal name at the command prompt; otherwise, the default logon name with the `admin` instance appended to the default logon name is used. If you specify the `-n` switch, the default logon name is used and the `admin` instance is not automatically appended to the logon name.

The `kadmin` command-line administrator uses the following methods to authenticate the administrator:

- The first method prompts administrators for a password.
- The second method uses the `-k` switch, which notifies `kadmin` to search the `v5srvtab` file for the key. With the `-k` switch, you can write shell scripts to automate administrative tasks. Read the permissions in the `v5srvtab` file to use this switch.

The communication between the `kadmin` client and the server daemon are encrypted to prevent disclosure of information across the network.

After you are authenticated, use the `kadmin` commands to manage the principal database. The `kadmin` commands are discussed in the subsequent sections of this chapter.

NOTE

You cannot use `kadmin` to control the following parameters of the user principals:

- Administrative permissions
- Default group prinicpal
- Maximum ticket lifetime and renew times
- Addition of new realms
- Alter key types

HP recommends that you use the graphical user interface administrative utility, `kadminl_ui`, to administer these parameters.

Adding a New Principal

You must specify the `add` administrative privilege in `admin_acl_file` to add a principal to the database.

To add a new principal, type `kadmin add` at the HP-UX prompt. This command adds a new principal with the specified name and password to the principal database. When you add a principal by using the `add` command, the principal inherits the default group principal settings for the key type and salt types.

The general syntax for adding a new principal is as follows:

```
command: add
```

NOTE You must specify values for all the mandatory LDAP attributes while creating a Kerberos principal. These attributes need to be specified only if the LDAP DN does not exist in the Directory server. You are prompted for mandatory attributes based on the default object class template that you specified while configuring your Kerberos server with LDAP as the backend. You are not prompted for LDAP attributes if the default object class template consists of only one mandatory attribute.

When creating principal names, ensure that a principal name meets the following conditions:

- Is case-sensitive.
- Is shorter than 767 characters.
- Is uniquely defined in the first 255 characters.
- Do not contain a space, tab, # (pound sign), \ (backslash) or : (colon).
- Does not subscribe to a NULL policy.
  If you subscribe to a policy that does not exist in the `password.policy` file, the default policy `*` is applied to the principal.

For example, to add a new principal admin, type kadmin at the HP-UX prompt, and specify the add command, the principal name, and the policy name. Following is a sample output of the add command:

```
command:  add
Name of Principal to add: admin
Enter password: <password>
Re-enter password for verification: <password>
Enter policy name (Press enter key to apply default policy):
Enter LDAP DN name (Press enter key to apply default policy):

Principal added
```

Adding a Random Key

Service principals use the addrnd command to add the specified principal with a randomly generated password.

The general syntax for adding a principal with a randomly generated password is as follows:

```
command: addrnd
```

For example, to add a principal admin with a randomly generated password, type kadmin at the HP-UX prompt, and specify the addrnd command, the principal name, and the policy name. Following is a sample output of the addrnd command:

```
command:  addrnd
Name of Principal to add: admin
Enter policy name (Press enter key to apply default policy):
Enter LDAP DN name (Press enter key to apply default policy):
Principal added
```

Specifying a New Password

The cpw command enables you to specify a new password for the principal.

The general syntax for specifying a new password is as follows:

```
command: cpw
```

For example, to enable a new password for the principal admin, type kadmin at the HP-UX prompt, and specify the cpw command and the principal name. Following is a sample output of the cpw command:

```
command: cpw
Name of Principal: admin
Enter password: password
Re-enter password for verification: password
Principal modified
```

Changing Password to a New Randomly Generated Password

The `cpwrnd` command changes the password of a principal to a new randomly generated password.

The general syntax for the `cpwrnd` command is as follows:

```
command: cpwrnd
```

For example, to change the password of the principal `admin` to a new randomly generated password, type `kadmin` at the HP-UX prompt, and specify the `cpwrnd` command and the principal name. Following is a sample output of the `cpwrnd` command:

```
command: cpwrnd
Name of Principal: admin
Principal modified
```

Deleting a Principal

To delete a principal, you must specify the `delete` administrative privilege in `admin_acl_file`.

The general syntax for deleting a principal is as follows:

```
command: del
```

For example, to delete the principal `admin`, type `kadmin` at the HP-UX prompt, and specify the `del` command and the principal name. Following is a sample output for the `del` command:

```
command: del
Name of Principal to delete: admin
Principal removed
```

The message `Principal removed` indicates that the principal is deleted.

Extracting a Principal

The `ext` command securely extracts the key of the principal into a local service key table file. By default, the `host/fqdn@REALM` principal is extracted into the `v5srvtab` file, where `fqdn` is the fully qualified domain name of the host system.

If the principal does not exist in the principal database, it is added with the name that you have specified. If the service key table file does not exist, it is created with the name that you have specified.

If the principal exists, `kadmin` resets the key version to 1 by overwriting the previous key and extracting the key that is created using a new password and no salt key. To extract the key without any modifications, use the `-n` option.

The general syntax for extracting a principal and a key to a local service key table file is as follows:

```
command: ext
```

For example, to extract the principal `admin` to a local service key table file, `SrvTab`, type `kadmin` at the HP-UX prompt and specify the `ext` command, the principal name, and the service key table file name. Following is a sample output for the `ext` command:

```
command: ext
Name of Principal (host/fqdn@REALM): admin
Service Key Table File Name (/opt/krb5/v5srvtab):/opt/SrvTab
Principal modified
Key extracted
```

The optional parameters are as follows:

`[-n]`	Extracts the key for an existing principal without changing the key or the salt type.
`[-p keytype]`	Defines the key type for the primary key, and extracts it to the service key table file. Supported values for `keytype` are `1` for DES-CRC, `3` for DES-MD5, and `5` for 3DES.
`[-a keytype]`	Defines the key type for the secondary key, and extracts it to the service key table file. Supported values for key type are `0` for no secondary key, `1` for DES-CRC, `3` for DES-MD5, and `5` for 3DES.

`[principal]`	Specifies an alternate principal to extract other than the default `host/fqdn@REALM` principal, for example, `ext finance@BAMBI.COM`

After `ext` executes, it prompts you for the service key table file name. The default file name is `/krb5/v5srvtab`.

Listing the Attributes of a Principal

The `inq` command lists the attributes of the principal, if it exists. The `kadmin` utility displays the following message if the principal does not exist in the database:

```
Principal does not exist
```

The general syntax for listing the attributes of the principal is as follows:

```
command: inq
```

For example, to list the attributes of the principal `admin`, type `kadmin` at the HP-UX prompt, and specify the `inq` command and the name of the principal. Following is a sample output for the `inq` command:

```
command: inq
Name of Principal: admin
```

Modifying a Principal

To modify attributes of a principal in the database, use the `kadmin mod` command. In addition to the name of the principal, you must also specify the type of parameter.

You can modify the following parameters:

`attr`	Specifies properties associated with the principal.
`fcnt`	Specifies the number of failed authentication attempts allowed for a principal. The number must be an integer between 0 and 255.
`vno`	Specifies the key version number. The number must be an integer between 0 and 255. When you create a principal, its key version number (`vno`) is 1 and the version number automatically increments by one each time the key is changed.

`policy`	Specifies the new policy name. If you do not specify a policy name, the default policy is applied.
`dn`	Specifies the LDAP DN name. If you do not specify an LDAP DN name, the default policy is applied.

The general syntax for modifying an existing principal is as follows:

```
command: mod
```

For example, to modify the principal `admin`, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, and the type of parameter. Following is a sample output for the `mod` command:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t):<parameter_type>
Principal modified.
```

Enter the *<parameter_type>* to be modified at the command prompt. The principal is modified depending on the parameter that you have specified. The subsequent sections of this chapter contain a detailed description of the parameter types.

Number of Authentication Failures (`fcnt`)

When you create a principal, the failed authentication count is automatically set to zero. The user associated with that principal increments the failed authentication count by 1 for each failed authentication attempt.

If the user has more consecutive authentication failures than allowed by the `MaxFailAuthCnt` parameter in the password policy file, the principal is `locked`. Before the user can attempt to authenticate again, the administrator must unlock the principal, which resets the `fcnt` to zero.

If the user successfully authenticates before the maximum failed authentication count value, `fcnt` is automatically reset to zero.

For example, to modify the `fcnt` parameter for the principal `admin`, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, and the `fcnt` parameter.

Following is a sample output for the `mod` command with the `fcnt` parameter:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno,policy,dn or quit
):fcnt
Failure Count (or quit): <enter count>
Principal modified.
```

Key Version Number Attribute

Every principal password has an associated version number that identifies the frequency of password changes.

When you create a principal, its password version number is inherited from the default group template. Every time you change a password, the version number is incremented by 1. However, the key version number is a writable field in the database, and you cannot consider the key version number as an accurate counter.

If you change the password for the default group principal, the key version number increments, and every new principal added to the database does not begin with 1. Because this may be confusing, HP recommends that you edit the version number for the default group principal to be 1 after you regenerate a key for the default principal.

For example, to modify the `vno` parameter for the principal `admin`, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, and the `vno` parameter.

Following is a sample output for the `mod` command with the `vno` parameter:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno,policy,dn or quit
): vno
Remote Administrator version Number (or quit): <enter count>
Principal modified.
```

LDAP DN

This option specifies the LDAP DN name. The LDAP DN specifies the location of where all information for the client is found in the DIT. If you choose to use the default, press the return key.

For example, to modify the `dn` parameter for the principal `admin`, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, and the `dn` parameter.

Following is a sample output for the mod command with the dn parameter:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :dn
Enter LDAP DN name or quit: <enter LDAP DN name>
Principal modified.
```

Policy Name

This option specifies the policy name subscribed by the principal. When the principal attempts to change its password, the rules of this policy are applied to the principal.

For example, to modify the policy parameter for the principal admin, type kadmin at the HP-UX prompt and specify the mod command, the principal name, and the policy parameter.

Following is a sample output for the mod command with the policy parameter:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :policy
Enter policy name or quit: <enter policy name>
Principal modified.
```

Attributes

Attributes indicate the properties associated with a principal. To view a list of attributes, type kadmin at the HP-UX prompt and specify the attr command. Following is a sample output of the attr command:

```
Command: attr
Attribute (or quit): ?
```

The subsequent sections discuss the attributes associated with a principal.

Allow Postdated Attribute

The Allow Postdated attribute determines whether a principal is allowed ticket postdating. Postdating is a mechanism that allows a principal to obtain a ticket that is initially invalid, but that becomes valid in the future.

The Allow Postdated attribute applies to both user and service principals specified as follows:

- You can issue either a postdated or postdatable ticket for user principals.
- The server can issue postdated service tickets for the service.

NOTE Before the server issues a postdated service ticket, the requesting user must possess a postdatable TGT.

To modify the type of the parameter attr for the principal admin and to set the Allow Postdated attribute, type kadmin at the HP-UX prompt and specify the mod command, the principal name, the attr parameter type, and the attribute.

Following is a sample output of the Allow Postdated attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno,dn or quit) :attr
Attribute (or quit): {postdate|nopostdate}
Principal modified.
```

Allow Renewable Attribute

The Allow Renewable attribute determines whether a principal is allowed to request renewable tickets. Renewable tickets are those that can be revalidated up to the maximum renewal time.

The principal database krbtgt/REALM@REALM principal contains the maximum ticket lifetime and the renewable time. You can use the Maximum Renew Time Setting in the General tab of the Principal Information window to limit individual principal accounts.

The Allow Renewable attribute applies to both user and service principals. If this attribute is set to a user principal, the principal can be issued a renewable ticket. If this attribute is set to a service principal, the server can issue a renewable ticket for the service.

NOTE Before the server issues a renewable service ticket, the requesting user must possess a renewable TGT.

To modify the type of the parameter `attr` for the principal `admin` and to set the `Allow Renewable` attribute, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, the `attr` parameter type, and the attribute.

Following is a sample output of the `Allow Renewable` attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :attr
Attribute (or quit): {renew|norenew}
Principal modified.
```

Allow Forwardable Attribute

The `Allow Forwardable` attribute determines whether a principal is allowed ticket forwarding. Forwarding is a mechanism to send a TGT to a remote system, from one network host to another. On behalf of the principal, you can use the forwarded TGT to generate a new service ticket on the second host system. This eliminates the need for the user to reauthenticate and re-enter the password on the second host.

The `Allow Forwardable` attribute applies to both user and service principals. If this attribute is set for a user principal, you can issue a forwarded or forwardable ticket to the principal. If this attribute is set for a service principal, the server can issue a forwarded service ticket for the service.

NOTE Before the server issues a service ticket on the remote host, the requesting user must possess a forwarded TGT.

To modify the type of the parameter `attr` for the principal `admin` and to set the `Allow Forwardable` attribute, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, the `attr` parameter type, and the attribute.

Following is a sample output of the `Allow Forwardable` attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :attr
Attribute (or quit): {forward|noforward}
Principal modified.
```

Allow Proxy Attribute

The `Allow Proxy` attribute determines whether a principal is allowed proxy tickets. Proxy tickets allow applications that a principal accesses with a TGT to request a special class of service ticket. You can move this type of service to another host on the network that acts on behalf of the principal, for example, a print service printing a file for a user.

NOTE You can use the authorization fields of the ticket and authenticator to hold restrictions on the proxy ticket.

The `Allow Proxy` attribute applies to both user and service principals. If this attribute is set for a user principal, you can issue a proxy ticket to the principal. If this attribute is set for service principal, the server can issue a proxy service ticket for the service.

NOTE Before the server issues a proxy service ticket, the requesting user must possess a proxy TGT.

To modify the type of parameter `attr` for the principal `admin` and to set the `Allow Proxy` attribute, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, the `attr` parameter type, and the attribute.

Following is a sample output of the `Allow Proxy` attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :attr
Attribute (or quit): {proxy|noproxy}
Principal modified.
```

Allow Duplicate Session Key Attribute

The `Allow Duplicate Session Key` attribute determines whether a principal is allowed to use a duplicate session key. A duplicate session key applies to user-to-user authentication and determines which key is used to encrypt the requested service tickets.

This setting controls the security protocol between a client application, `initiator`, and a service called the `acceptor`. The following processes occur when an initiator application requests for a duplicate session key:

- The initiator application sends the TGT of the initiator and the acceptor as a request to the ticket-granting service (TGS). The `Allow Duplicate Session Key` attribute is set. The service ticket returned to the initiator is encrypted with the session key in the TGT of the acceptor.
- The service ticket returned to the initiator application is encrypted with the secret key of the acceptor when the `Allow Duplicate Session Key` attribute is not set.

This attribute is set by default, thereby allowing an initiator application to request a duplicate session key for the application of the acceptor. You must assign the `Allow as Service` attribute to principal accounts that use duplicate session keys.

To modify the type of parameter `attr` for the principal `admin` and to set the `Allow Duplicate Session Key` attribute, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, the `attr` parameter type, and the attribute.

Following is a sample output of the `Allow Duplicate Session Key` attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :attr
Attribute (or quit): {dskey|nodskey}
Principal modified.
```

Require Preauthentication Attribute

The `Require Preauthentication` attribute determines whether a principal is required to preauthenticate when requesting a TGT. Preauthentication implies that the client logon program attaches known encrypted data to a ticket request, providing additional security when the TGT is presented to access a secured service.

The `Require Preauthentication` attribute applies to user and service principals. If this attribute is set for a user principal, the user must run the logon software that performs authentication using the preauthentication protocol. If this attribute is set for a service principal, the service accepts TGTs only from user principals that obtained a TGT using a preauthentication protocol.

NOTE Client applications require preauthentication by default; however, a client can override this setting.

To modify the type of parameter `attr` for the principal `admin` and to set the `Require Preauthentication` attribute, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, the `attr` parameter type, and the attribute.

Following is a sample output of the `Require Preauthentication` attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :attr
Attribute (or quit): {preauth|nopreauth}
Principal modified.
```

Require Password Change Attribute

The `Require Password Change` attribute determines whether a principal must change the password of the user during the next authentication attempt. You must change the password when this attribute is set for a principal.

When a new principal is added to the database or when a password of the principal is changed, this attribute is controlled by the NoReqChangePwd setting in the password policy file of the principle. By default, NoReqChangePwd is set to 0 (zero), that is, the user must change the password at first logon.

If you designate a random key for a principal using the HP Kerberos Administrator window or the kadmin addrnd command, the Require Change Password attribute is not set by default. As a result, a service principal with an extracted key need not contain a new key extracted during the next authentication attempt.

To modify the type of parameter attr for the principal admin and to set the Require Password Change attribute, type kadmin at the HP-UX prompt and specify the mod command, the principal name, the attr parameter type, and the attribute.

Following is a sample output of the Require Password Change attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :attr
Attribute (or quit): {pwchg|nopwchg}
Principal modified.
```

Lock Principal Attribute

The Lock Principal attribute determines whether a principal account is usable or not. A locked principal exists in the principal database but is unable to use or provide security network services.

The Lock Principal attribute applies to both user and service principals. If you set this attribute for a user principal, no tickets can be issued to the user. If you set this attribute for a service principal, no tickets are issued for principals to use the service.

This attribute is set automatically when a principal exceeds the maximum number of failed authentication attempts specified in the password policy file. The default maximum number of failed authentication attempts allowed is 5. If a principal account is locked, a principal with the required administrative permissions must unlock the principal account before the user can authenticate again.

To modify the type of the parameter `attr` for the principal `admin` and to set the `Lock Principal` attribute, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, the `attr` parameter type, and the attribute.

Following is a sample output of the `Lock Principal` attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :attr
Attribute (or quit): {lock|nolock}
Principal modified.
```

Allow As Service Attribute

You must select the `Allow As Service` attribute for any principal that is used as a service.

This attribute applies to both the user principal and the service principal. Selecting this attribute does not necessarily mean that the principal account is being used by a network service application. Select this attribute for user principals that run programs requiring user-to-user authentication.

When you set the `Allow As Service` attribute, the name of the principal name appears in the server field of the service ticket. If you do not set this attribute, the Kerberos server cannot issue a service ticket for that principal because the name of the principal cannot appear in the server field of the service ticket.

This attribute is set by default, allowing principals to act as a service and enabling user-to-user authentication for user principals.

To modify the type of the parameter `attr` for the principal `admin` and to set the `Allow As Service` attribute, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, the `attr` parameter type, and the attribute.

Following is a sample output of the `Allow As Service` attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :attr
Attribute (or quit): {svr|nosvr}
Principal modified.
```

Require Initial Authentication Attribute

The `Require Initial Authentication` attribute specifies if the server is allowed to issue service tickets to a service principal on behalf of a user principal using an existing TGT.

The `Require Initial Authentication` attribute applies only to service principals. If you set this attribute, user principals must reauthenticate to the Kerberos server before the server issues a service ticket for that service. For example, the change password service requires a principal to enter a password to receive a ticket for the change password service before changing the password. If you set this attribute, the server may issue a service ticket based on the existing TGT of the user principal.

NOTE

In **Principal Information>Edit>Edit Administrative Permissions**, if you select the `Require Initial Authentication` attribute, the `Allow as Service Attribute` is automatically selected.

Do not enable this setting for user principal accounts. This attribute is applicable to selected service principals.

To modify the type of parameter `attr` for the principal `admin` and to set the `Require Initial Authentication` attribute, type `kadmin` at the HP-UX prompt and specify the `mod` command, the principal name, the `attr` parameter type, and the attribute.

Following is a sample output of the `Require Initial Authentication` attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :attr
Attribute (or quit): {tgt|notgt}
Principal modified.
```

The `notgt` command in `kadmin` is equivalent to selecting the `Require Initial Authentication` in the `tgt` command in `kadmin` is equivalent to clearing the `Require Initial Authentication` checkbox on the Principal Information window>Attributes tab.

You can use the kadmin inq command to view the attribute of the principal. With Require Initial Authentication selected (tgt), the inquire command shows TGT_BASED in the attributes field. Without the Require Initial Authentication setting (notgt), the text does not appear in the attributes field.

Table 8-18 displays the output of the HP Kerberos Administrator>Attributes tab setting that is equivalent to the kadmin command. It also indicates the attribute text that displays when you view the setting of the principal using the kadmin inq command.

Table 8-18 Require Initial Authentication Attribute Settings

Attributes Tab Check-Box Setting	HP Kerberos Administrator	kadmin inq Shows:
Select Require Initial Authentication	notgt	No text shows
Select Require Initial Authentication	tgt	TGT_BASED

Set As Password Change Service Attribute

The Set As Password Change Service attribute determines if a service principal can act as a change password service. If you set this attribute, a service principal receives initial tickets for user principals whose passwords have expired.

NOTE When you select the Set As Password Change Service attribute, the Require Initial Authentication attribute is automatically selected.

Normally, you select the Set As Password Change Service attribute only for the service principal defined as a change password service. You can add other change password service principals to the principal database if you have created custom applications that require different password service principals.

To modify the type of parameter attr for the principal admin and to set the Password Change Service attribute, type kadmin at the HP-UX prompt and specify the mod command, the principal name, the attr parameter type, and the attribute.

Following is a sample output of the `Password Change Service` attribute:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr, fcnt, vno, policy,dn or q
ui) :attr
Attribute (or quit): {cpwsrv|nocpwsrv}
Principal modified.
```

Password Expiration Attribute

A principal password may have a finite or an infinite lifetime. Following are the factors that control the expiration time of a password, including the principal type:

- Service Principals – The secret key stored in the service key table file on the host of the service does not expire. However, HP recommends that you extract new random keys periodically for best security practices. See “Maintaining Secret Keys in the Key Table File” on page 244, for more information.
- User principals – The expiration time for the password of a user depends on the settings designated for the principal account.

  Activating the `Password Expiration` attribute holds a principal in accordance with the password expiration policy. You are prompted to change the password before the expiration date. If you do not enable the `Password Expiration` attribute, the password of the current principals never expires.

NOTE

The password expiration date is stored on the Kerberos server with each principal. When you change the password, the current date and the expiration value also change in the password policy file.

Before the password expires, you are notified that the password is about to expire. The `NotifyTime` parameter controls the advance notice timing in the password policy file. If you ignore the advance notice and the expiration date elapses, you must change the password before you can obtain any more tickets from the Kerberos server.

Because the expiration time is calculated from the time you add a new principal to the database, the password change load on the server is distributed over time. Therefore, you can select a password expiration in the default group principal template without affecting the administrative load, provided you add new principals over a period of time.

To modify the parameter type `attr` of the principal `admin` to set the `Password Expiration` attribute, you need to execute the following:

```
Command: mod
Name of Principal to Modify: admin
Parameter Type to be Modified (attr,fcnt,vno, policy,dn or qui
t) :attr
Attribute (or quit): {cpwexp|nocpwexp}
Principal modified.
```

Principal Expiration Attribute

The `Principal Expiration` attribute determines the expiration time of a principal account. You can set the expiration time to a definite time or to never. An expired principal account is essentially locked; it can no longer be used to access the security network. However, this account can be re-enabled by resetting the expiration time, because the principal still exists in the principal database.

Setting a principal expiration time may be useful for granting access to temporary employees. However, if you specify an expiration date for the default group principal, all principals added using that template setting will expire at the same time. You must consider the administrative requirements of expiring all principal accounts on the same day.

You cannot set this attribute using the command-line administrator.

Maximum Ticket Lifetime Attribute

The `Maximum Ticket Lifetime` attribute determines the maximum lifetime for an initial or service ticket that the principal requests. If you set the lifetime to a time longer than the lifetime assigned to the `krbtgt/REALM@REALM` principal, the settings in the `krbtgt/ principal` take precedence.

You may choose to set a maximum ticket lifetime for the default group template that is different from the `krbtgt/ principal` if you plan to enter a block of users that have restricted ticket lifetimes. After adding the block of user principals, you can alter the default group setting again.

You cannot set this attribute using the command-line administrator.

Maximum Renew Time Attribute

The `Maximum Renew Time` attribute controls the renew time limit for renewable tickets. If you set the renew time longer than the renew time assigned to the `krbtgt/REALM@REALM` principal, the settings in the `krbtgt/ principal` take precedence.

You cannot set this attribute using the command-line administrator.

Key Type Attribute

The `Key Type` attribute generates a secret key. You must carefully decide which encryption type to use for the secret key.

Each principal can be associated with two different secret keys. These are called the primary and secondary keys. Each key is associated with an encryption type. The encryption type designates the encryption algorithm used to generate the secret key. Following are the supported encryption types:

- DES3 (Security-Enhanced Triple Data Encryption Standard)
- DES-MD5
- DES-CRC

You cannot set this attribute using the command-line administrator.

Salt Type Attribute

A salt is a string of characters added to the beginning of a password before the password is transformed into a secret key. Salts strengthen passwords and ensure that principals with the same password do not have the same key. Salt settings apply only to user principals; service principals use a random key, and they do not require a designated salt (they use a salt type of `None`).

You can control the salt type attributes using the Password tab in the HP Kerberos Administrator Principal Information window.

You cannot set this attribute using the command-line administrator.

Principal Database Utilities

Principal database utilities are tools that help you to globally manage the principal database. Use these tools only if the database was not properly created or configured during installation, or if you are debugging or upgrading your Kerberos server.

To use the principal database utilities, you must have `root` user privileges. Table 8-6 lists the global database tasks and the tools used to perform each task.

Table 8-19 Principal Database Utilities

Utility	Task
`kdb_create`	Create a principal database.
`kdb_destroy`	Securely remove the principal database.
`kdb_dump`	Dump the principal database to `stdout` or to a specified text file.
`kdb_load`	Load all principal names into the principal database from an existing database dump file.
`kdb_stash`	Stash the master key of the Kerberos server.

Principal database utilities are described in detail later in this chapter. Export the following path before executing any of the commands at the HP-UX command prompt:

```
$ export PATH=$PATH:/opt/krb5/admin
```

Kerberos Database Utilities

The primary security server contains a database of all principals that are trusted in each of the supported realms. You can also create the database during installation. See “Auto-Configuration of the Kerberos Server” on page 63 for more information.

The `kdb_create` utility creates a Kerberos database and adds a realm to the existing database. You cannot use this utility if you do not remember the master password. After creating the principal database using the `kdb_create` utility, you can load a previously dumped database by using the `kdb_load` utility.

NOTE You must be a `root` user to execute the `kdb_create` utility.

The general syntax for creating the Kerberos database is as follows:

```
kdb_create [-a REALM] [-e enctype] [-M mkeyname] [-p
PASSWORD] [-r REALM] [-s[-f keyfile]] [-v]
```

The `-a`, `-e` and the `-M` switches are used to override defaults. You must use these switches each time you run other daemons and programs that use the defaults. For example, when you use the `kadmind` or `kdb_load` utility, use the switches mentioned in the `kdb_create` command.

Restart the `kadmind` and the `kdcd` daemons after you invoke the `kdb_create` utility.

You can invoke the `kdb_create` utility with the following options:

`-a Realms` Adds the realm `REALM` to the existing principal database. To use this switch, you must be aware of the master password and the principal database must already exist.

`-e enctype` Specifies the encryption and checksum mechanism of the primary principal. Following are the encryption types that are supported:

- `3DES` or `5`: `DES-CBC-MD5` (default)
- `DES-MD5` or `3`: `DES-CBC-MD5`

- DES-CRC or 1: DES-CBC-CRC

NOTE

The default, DES3-CBC-MD5, will be set as the encryption type if you do not specify any of the encryption types previously mentioned.

-f keyfile	Specifies an alternate name for the stash file when used with the -s switch. If you do not use the -f switch, .k5.REALM is used as the default keyfile.
-M mkeyname	Specifies an alternate primary principal name. The default primary name is K/M@REALM.
-p PASSWORD	Suppresses kdb_create from prompting you for the master password, which makes it easier to configure a database with a shell script. Uses the master password to generate an encryption key that protects all the entries in the database. You cannot use this option to change the master password.
-r REALM	Creates the principal database for the realm REALM. By default, kdb_create uses the realm defined in the krb.conf file. If this file does not exist, the command uses the uppercase equivalent of the domain name.
-s	Stores the master key in a stash file that can be automatically retrieved, eliminating the need to manually enter the key each time you start the Kerberos server.
-v	Runs kdb_create in verbose mode.

The following example displays how to use kdb_create:

```
shell% kdb_create -a BAMBI.COM
Initializing database /opt/krb5/principal for realm BAMBI.COM.
   master key name is K/M@DCETST3.FINANCE.BAMBI.COM

You will be prompted for the database Master Password
It is important that you DO NOT FORGET this password.
Enter password:
Re-enter password for verification:
```

```
Adding principals to database...
Cleaning up....
shell%
```

The kdb_create command creates the following principals:

- K/M@<REALM NAME>
  This is the default key name. However, you can configure this key name.
- default@<REALM NAME>
- kadmin/<REALM NAME>@<REALM NAME>
- kcpwd/<REALM NAME>@<REALM NAME>
- krbtgt/<REALM NAME>@<REALM NAME>

IMPORTANT Do not delete these principals.

The K/M keyname is the default master key name. However, you can change the master key name by specifying the tag while using the -M mkeyname option in kdb_create command.

The stash file is a local copy of the master key that resides on the local disk of the primary security server in an encrypted format. This stash file is usually located in the same directory as the Kerberos database. By default, kdb_create does not create a stash file. A stash file allows the database utilities, such as kadmind, kadminl, kdcd and others, to authenticate themselves.

Occasionally, however, you may have to restart the machine on which the KDC runs, and if a stash file is present, you can configure KDC to start automatically without any human intervention whenever the machine is rebooted. The stash file, like the keytab file, is a potential point-of-entry for a break-in, and if compromised, allows unrestricted access to the Kerberos database. For more information, see “Service Key Table” on page 244.

Database Encryption

The Kerberos server supports the following encryption types:

- DES3

- DES-MD5
- DES-CRC

The encryption type selected during database creation determines the encryption type applied to the master password, which in turn is used to create the key that secures all records stored in the principal database.

Encrypt the database using DES encryption if you are installing a secondary security server that has an existing principal database encrypted using DES. In this case, do not create the database during installation. Instead, use the `kdb_create` utility to create the database after installation.

Regardless of the database encryption choice, the installation program always installs both DES and 3DES algorithms. Therefore, you can specify any key type for individual principal accounts in the database.

Database Master Password

When you create the principal database, you must supply a master password. The master password, along with the specified encryption type, generates the master key that protects the database entries. In other words, the stored keys of each principal account are encrypted with the master key. This provides double security protection for each stored key.

The `kdb_create` utility prompts you for the master key for the Kerberos database. This key can be any string. A good key is one you can remember, but that no one else can guess. Examples of bad keys are words that can be found in a dictionary; any common or popular name, especially a famous person or a cartoon character; or your user name in any form (forward, backward, repeated twice, and so on).

Destroying the Kerberos Database

The kdb_destroy utility securely removes the principal database. This utility runs on the primary and secondary security servers. If you run this utility using command-line options, it prompts you with a confirmation message and then removes the default principal database, /krb5/prinicpal. To confirm the deletion, type **yes** otherwise, kdb_destroy returns the message Database not destroyed.

This tool destroys only the principal.* files. You must handle the other files that store the principal information separately. To destroy admin_acl_file, manually delete it. To destroy the key table files, use the ktutil tool.

To ensure that no one reads the previous contents of the database files, kdb_destroy writes 0s (zeros) to the original files before it deletes them.

The general syntax for destroying the Kerberos database is as follows:

```
kdb_destroy [-f keyfile]
```

The kdb_destroy utility uses the following options:

-f keyfile	Destroys an alternative key file named keyfile.
-e enctype	Specifies the encryption and checksum mechanism of the primary principal. Following are the encryption types that are supported: • 3DES or 5: DES-CBC-MD5 (default) • DES-MD5 or 3: DES-CBC-MD5 • DES-CRC or 1: DES-CBC-CRC

NOTE

The default, DES3-CBC-MD5, will be set as the encryption type if you do not specify any of the encryption types previously mentioned.

Following is an example output of the kdb_destroy utility:

```
shell% kdb_destroy
keyfile: /opt/krb5/.k5.DCETST3.FINANCE.BAMBI.COM
Deleting KDC database stored in '/opt/krb5/principal', are you
```

```
 sure?
(type 'yes' to confirm)?
Database destroyed!
```

Dumping the Kerberos Database

The kdb_dump utility copies the contents of the principal database to stdout or to a text file. By default, the output is displayed on the terminal using the stdout command.

NOTE You must be a root user to run the kdb_dump program.

The general syntax for dumping the Kerberos database is as follows:

```
kdb_dump [-f filename]
```

The kdb_dump utility uses the following option:

`-f filename` Specifies a destination file name and path to store the output. If you do not use the -f switch, the text is sent to stdout.

`-e enctype` Specifies the encryption and checksum mechanism of the primary principal. Following are the encryption types that are supported:

- 3DES or 5: DES-CBC-MD5 (default)
- DES-MD5 or 3: DES-CBC-MD5
- DES-CRC or 1: DES-CBC-CRC

NOTE The default, DES3-CBC-MD5, will be set as the encryption type if you do not specify any of the encryption types previously mentioned.

Following is an example of using kdb_dump:

```
shell% kdb_dump -f <filename>
```

Loading the Kerberos Database

The kdb_load utility loads the database with the principal entries from a database dump text file. This utility overrides the existing database entries with the corresponding entries present in the dump file. Principals in the existing database that are absent in the dump file are not changed or removed.

HP recommends that you run the kdb_load utility on the primary security servers only. If the propagation is configured correctly, kdb_load securely copies the database entries to each secondary security server present on your network.

The general syntax for loading the Kerberos database is as follows:

```
kdb_load [-f filename]
```

The kdb_dump utility uses the following options:

`-f filename` Specifies the database dump text file to copy into the database.

`-e enctype` Specifies the encryption and checksum mechanism of the primary principal. Following are the encryption types that are supported:

- 3DES or 5: DES-CBC-MD5 (default)
- DES-MD5 or 3: DES-CBC-MD5
- DES-CRC or 1: DES-CBC-CRC

NOTE The default, DES3-CBC-MD5, will be set as the encryption type if you do not specify any of the encryption types previously mentioned.

Following is an example of using kdb_load:

```
shell% kdb_dump -f <filename>
```

Stashing the Master Key

The kdb_stash utility stores the master key, the encrypted master password, to a stash file. This utility runs on the primary and secondary security servers. Use the kdb_stash utility to store the master key in a stash file. You must specify the same key type and master password that you specified when you created the database.

NOTE If you have used the kdb_create -s utility while creating your database, you already have a stash file.

If you store the password in a disk file, it may allow an intruder to gain access to the principal database. Therefore, secure the file carefully.

The general syntax for stashing the master key is as follows:

```
kdb_stash [-e enctype] [-f keyfile] [-M mkeyname] [-r REALM]
```

The kdb_stash utility uses the following options:

`-e enctype` Specifies the encryption type to be used to generate the master key. The type you specify must be the same as the type you specified while creating the database. Following are the encryption types that are supported:

- 3DES or 5: DES-CBC-MD5 (default)
- DES-MD5 or 3: DES-CBC-MD5
- DES-CRC or 1: DES-CBC-CRC

`-f keyfile` Stashes the key in an alternate key file named keyfile. If you do not use the -f switch, the default keyfile is .k5.REALM.

`-M mkeyname` Specifies an alternate for the primary principal name. The default primary principal name is K/M@REALM.

`-r REALM` Stashes the principal database key for the realm REALM. By default, kdb_stash uses the realm defined in the krb.conf file. If the file does not exist, the command uses the uppercase equivalent of the domain name.

Following is an example of using `kdb_stash`:

```
shell% kdb_stash -f <filename>
Enter password: <password>
Re-enter password for verification: <password>
```

Starting and Stopping Daemons

If you change the configuration of the Kerberos server, you must stop and restart the services and daemons for the changes to take effect. Table 8-8 briefly describes the related services and daemons that you must stop and restart.

Table 8-20 Starting and Stopping Daemons and Services

Situation	Daemons and Services
Propagation startup or changes	`kadmind`, `kdcd`, and `kpropd` daemons
Configuration changes	`kadmind`, `kdcd`, and `kpropd` daemons

You can use the `krbsrv` tool to start and stop the following Kerberos daemons:

- `kdcd`
- `kadmind`

NOTE You can also use the `krbsetup` tool to start and stop the Kerberos daemons.

You can use the `krbsrv` tool to start the `kdcd` and `kadmind` daemons by executing the following command at the HP-UX command prompt:

```
# /sbin/init.d/krbsrv start
```

You can use the `krbsrv` tool to stop the `kdcd` and `kadmind` daemons by executing the following command at the HP-UX command prompt:

```
# /sbin/init.d/krbsrv stop
```

Maintenance Tasks

Following are the maintenance tasks associated with the Kerberos server:

- "Protecting Security Server Secrets" on page 236
- "Backing Up primary security server Data" on page 237

Protecting Security Server Secrets

The Kerberos server stores the following types of secrets:

- `host/fqdn@REALM` service principal
- Master password

It is crucial that these secrets not be compromised. Performing simple maintenance tasks and following password protection guidelines help prevent security breaches.

host/fqdn@REALM

You require the `host/fqdn@REALM` service principal name for database propagation. You must change this key by generating a new key, extracting it to the server's service key table file, and deleting the old key. See "Maintaining Secret Keys in the Key Table File" on page 244, for more information on performing these tasks.

NOTE During key generation and extraction of the `host/fqdn@REALM` principal, the current service tickets become invalid; because service tickets are created at each application logon, application users are not affected by the update.

Master Password

You must enter the master password when installing a Kerberos server and when using the principal database utilities. You must select a strong password and make sure that it is kept safe from intruders. See "Database Master Password" on page 228, for more information on selecting and protecting the master password.

Backing Up primary security server Data

Save the copied information to a CD or tape — whatever your preferred archive method is.

Be aware that primary security server files contain sensitive information; therefore, do not copy files unless you intend to properly secure the backup copies.

Be sure to make backup copies of the following:

- `admin_acl_file`
- `password.policy` (`password.pol`)
- Principal database files
- `krb.conf`

Certain files contain extremely sensitive information, and HP recommends that you do not make backup copies of the following files:

- `.k5.REALM` — Instead, recreate this file by using the `kdb_stash` utility. You must know the master password and specify the correct encryption type to run this utility.
- `v5srvtab` — Instead, recreate this file by re-extracting the key for any service principal contained in the file — Typically, the `host/principal` for the primary security server.

Backing Up the Principal Database

If you have a server architecture that uses a second level of propagation servers, you can back up your principal database with minimal effect on application users. See Chapter 9, “Propagating the Kerberos Server,” on page 241.

NOTE If you do not use secondary security servers as propagation servers, you can temporarily halt propagation to one of the secondary security servers acting as an authentication server, provided you have a properly configured redundant server.

To back up your principal database, complete the following steps:

Step 1. Stop the services and daemons.

- Run the following command as a `root` user:

```
# /sbin/init.d/krbsrv stop
```

Step 2. Copy the `principal.dat`, `principal.idx`, and `principal.ok` files from one of the propagation servers to your desired destination, for example, CD-ROM or tape. The files are located at `/opt/krb5`.

Step 3. Restart the services and daemons by running the following command as a a `root` user:

```
# /opt/krb5/sbin/kpropd
```

All the new principal database information recorded on the primary security server during your database backup is copied to each secondary security server when propagation resumes.

Removing Unused Space from the Database

After long and continued use, the principal database on the primary security server can grow large due to unused space. When you delete a principal, the space that the record had occupied is not removed. Instead, the space is reserved and marked as available. Therefore, after extensive use, the database can grow very large. You can correct this by loading all existing principals into a new database.

To create a new database and load existing records on the primary security server, log on as a root user and complete the following steps:

Step 1. Stop the services and daemons by executing the following command:

```
# /sbin/init.d/krbsrv stop
```

Step 2. Make an archive of the `principal.*` files found on the primary security server.

Step 3. Dump the database by running the following command at the HP-UX command prompt:

```
# /opt/krb5/admin/kdb_dump -f /tmp/filename
```

Step 4. Delete the old database by running the following command at the HP-UX command prompt:

```
# rm /opt/krb5/principal*
```

Step 5. Create a new database using the same master password and encryption type as when the database was originally created by executing the following command:

```
# /opt/krb5/admin/kdb_create -s -e enctype
```

Step 6. Load the dump file by running the following command at the HP-UX command prompt:

```
# /opt/krb5/admin/kdb_load -f /tmp/filename
```

Step 7. Restart the services and daemons by running the following command at the HP-UX command prompt:

```
/sbin/init.d/krbsrv start
```

Step 8. Remove the `/tmp/filename` file after you have verified that the new database is functioning without problems.

9 Propagating the Kerberos Server

This chapter describes how to propagate the Kerberos database from the primary security server to the secondary security server.

This chapter discusses the following topics:

- “Propagation Hierarchy” on page 243
- “Service Key Table” on page 244
- “Propagation Tools” on page 246
- “The kpropd Daemon” on page 248
- “The mkpropcf Tool” on page 249
- “The kpropd.ini File” on page 251
- “The prpadmin Administrative Application” on page 257
- “Setting Up Propagation” on page 258
- “Monitoring Propagation” on page 263
- “Configuring Multirealm Enterprises” on page 272

NOTE Propagation is disabled if you select LDAP as your backend database. Check with your LDAP administrator, for more information about propagation of information on the LDAP Server.

Propagation Hierarchy

To authenticate users on the network, each secondary security server must contain the latest copy of the principal database, at all times. secondary security servers obtain the copy of the principal database from the primary security server using the database propagation service.

At predefined intervals, the database propagation service automatically copies database changes from the primary security server to its associated secondary security servers. The default propagation interval is 15 seconds.

A secondary security server acting as a propagation server can have other secondary security servers associated with it in a hierarchical configuration. In this case, a secondary security server that receives a copy of database changes from a primary security server through propagation must propagate those changes to other secondary security servers.

Propagation Relationships

You can define the relationship in a propagation hierarchy in the `kpropd.ini` file by assigning parent and child labels to servers. A parent server is a primary or secondary security server that propagates its database to another server. A child server is a secondary security server that receives the propagated database from another server. A child server may also act as a parent server for a different secondary security server.

Service Key Table

The `/krb5/v5srvtab` file is the service key table file that contains service principal names with their corresponding secret keys. You must store this file on the system that hosts the service or application, which requires an extracted key. Secured application servers use the keys in this file to decrypt data packets, which the security server encrypts, using a copy of the same key.

Maintaining Secret Keys in the Key Table File

Secret keys for service principals are randomly generated keys stored in the service key table on the host of the service principal. Periodically, you must change the secret keys for many service principals and delete the old keys. This requires generating a new random key, extracting the new key to the service key table file on the host of the service, and deleting the older keys. HP recommends that you perform these processes at least once a month. This reduces the risk of compromising the security of the keys.

Extracting a Key to the Service Key Table File

Only a principal whose account has the required administrative permissions can extract the keys. To extract a key to the service key table file on the host of the service, the principal must log on to the host system where the service resides and use the Administrator or the command-line administrator.

To extract a key to the service key table file using the Administrator, complete the following steps:

Step 1. Select the principal for which you want to extract the key.

Step 2. Click **Edit**. The Principal Information window displays.

Step 3. Select the **Edit>Extract To Service Key Table** option. The Extract to Service Key Table Window displays.

For more information on extracting a key to the service key table file, see “Extracting Service Keys” on page 178.

To extract the principal <*principal_name*> to a local service key table file, `SrvTab`, type `kadmin` at the HP-UX prompt and specify the `ext` command, the principal name, and the service key table file name.

Following is a sample output for the `ext` command:

```
command: ext
Name of Principal (host/fqdn@REALM): <Principal Name>
Service Key Table File Name (/opt/krb5/v5srvtab): <SrvTab>
Principal modified
Key extracted
```

Creating a New Service Key Table File

Each secured daemon requires a service principal account. You must extract the key of the principal to the service key table file. When you create a new service key table file, you must consider the number of daemons that reside on the system.

When you are creating a new service key table file, consider the following:

- Ensure that a single key table file is readable only by one user account. Do not set the read-write-execute permissions to a group or world.
- For a `host/principal`, you must use the default key table name, `/opt/krb5/v5srvtab`, and this must be owned by the root user.
- If some secured daemons on a single system run under the same UNIX® account, you can store more than one key in a given key table file.
- If secured daemons on one system run as more than one UNIX account, you must create one key table file for each UNIX account used by one of the secured daemons on the local system. To do this, use the `ktutil` command.

  For more information on the `ktutil` command, type `man 1 ktutil` at the HP-UX prompt.

Deleting Older Keys from the Service Key Table File

To remove principal entries from the service key table file, use the `ktutil` command. For more information on the `ktutil` command, type `man 1 ktutil` at the HP-UX prompt.

Propagation Tools

The kpropd daemon manages and performs propagation of the principal database on each server in the propagation hierarchy. It uses the following local files:

- prop_q

  A default propagation input queue file that contains the names of every principal whose record has changed since the last successful database propagation.

- prop_q.wrk

  A temporary working copy of prop_q, the default propagation input queue file.

- prop_hostname

  An ASCII cache file created on the propagating security server when the host name of the security server is unavailable during the principal database propagation cycle. The file contains principal records that were not propagated from the prop_q file to the receiving server.

- prop_hostname.ok

  An empty file created on the propagating security server after a successful propagation to the host name security server.

These files are critical and you must not compromise them under any circumstances.

Table 9-1 describes the propagation configuration tools.

Table 9-1 Propagation Tools

If You Want To:	Use This Tool:
Store the propagation configuration information used for propagation.	kpropd.ini – This is a text file.
Create the default kpropd.ini file, based on the contents of the configuration file, krb.conf.	mkpropcf

Table 9-1 Propagation Tools (Continued)

If You Want To:	Use This Tool:
Manually control propagation on one or more servers once propagation is configured and started.	`prpadmin`

For more information on the process for configuring propagation, see “Setting Up Propagation” on page 258.

This chapter contains a detailed discussion of these tools.

The kpropd Daemon

The /opt/krb5/sbin/kpropd daemon propagates the principal database from one server to another and starts running when the security server starts up. It propagates principal records from a given security server to kpropd on the receiving security server or to the propagation plug-in on the receiving security server, if kpropd is not running on this security system.

Propagation generally occurs downward through the propagation hierarchy from parent server to child server as configured in the kpropd.ini file.

During downward incremental propagation, kpropd refers to the prop_q.wrk file for changes to principal records and propagates only those records that have changed during the current propagation cycle.

When the failed authentication count of the principal increments, kpropd initiates upward propagation. During an upward incremental propagation, kpropd updates those principals on the primary security server whose failed authentication count values are incremented during the current propagation cycle. If propagation to a particular server fails, kpropd writes the unpropagated principal records to a prop_*hostname* file on the host name server.

At the end of a successful propagation, each security server has an up-to-date principal database, and each server above or below the propagating server in the hierarchy has an empty prop_hostname file, where *hostname* is the receiving server.

For a detailed description of propagation configuration, see “Setting Up Propagation” on page 258.

The mkpropcf Tool

The /opt/krb5/install/mkpropcf tool creates the kpropd.ini file, which is the default propagation configuration file in a propagation hierarchy. The mkpropcf tool exports the kpropd.ini file to the secondary security servers.

When you execute mkpropcf on the primary security server without any arguments, it creates the krpopd.ini file in the /opt/krb5 directory. The mkpropcf tool derives the information from the krb.conf file of the primary security server and takes into account only those security servers that are configured in the default realm, as specified in the krb.conf file.

NOTE

If the Kerberos configuration file, krb.conf, does not exist, mkpropcf creates a sample kpropd.ini file with 1 primary security server and 10 secondary security servers.

The general syntax for creating the kpropd.ini file is as follows:

```
/opt/krb5/install/mkpropcf [-d] [-e] [-i file_name] [-f]
```

The mkpropcf tool supports the following options:

-d	Deletes the existing the kpropd.ini file.
-e	Exports the information in the /opt/krb5/admin/kpropd.ini file into the temporary /opt/krb5/export.ini file on the primary security server. You must manually copy this file to the secondary security server.
-i	Imports the information from the temporary export.ini file into the kpropd.ini file of the secondary security server. If the configuration already exists, use the -f option to force a rewrite of the kpropd.ini file. If you do not use the -f option, mkpropcf displays an error message and does not create the new configuration file.

`-f`

Overwrites the kpropd.ini file. You can use this option with the `-i` option to explicitly overwrite the `kpropd.ini` file. To synchronize the `kpropd` configuration, HP recommends that you export the original configuration, `kpropd.ini`, on the primary security server to `export.ini`, and then copy the export file to each of the secondary security servers and execute the following command at the HP-UX prompt:

```
mkpropcf -i export.ini -f
```

The kpropd.ini File

The /opt/krb5/kpropd.ini file is the propagation configuration file created by the mkpropcf tool using the information from the local krb.conf file.

Ensure that only authorized users have access to this file. Unauthorized access to kpropd.ini can jeopardize the integrity of your realm. Intruders who modify or replace entries can also modify your principal database.

If you add or remove servers from the propagation hierarchy, that is, if you modify the kpropd.ini file, stop and restart the kpropd daemon on each security server. Stopping and restarting the kpropd daemon ensures that the servers correctly propagate to any new server added and do not propagate to the servers removed from the kpropd.ini file.

The general syntax for the kpropd.ini file is as follows:

```
[default_values]
interval=n[s|m|h|d]
key_exp=n[s|m|h|d]
max_cache=n[K|M]
max_retry_delay=n[s|m|h|d]
net_timeout=n[s|m|h|d]
port=port_name
primay_realm=DEFAULT_REALM
realms=[all|realm1[realm2][,...]]
service_name=service_principal_name
[secsrv1_name]
child=secsrv2_name
[secsrv2_name]
child1=secsrv3_name
child2=secsrv4_name
parent=secsrv1_name
```

When adding entries in the kpropd.ini file, consider the following:

- Specify values with a statement of the following type:

  ```
  key_phrase = value
  ```

- Any character following a pound sign (#) on a given line is ignored as comments. Blank lines are ignored.
- Use a backslash (\) to specify a line extension.

Sections

The kpropd.ini file stores configuration parameters required for propagation. This file contains the following sections:

- The [default_values] section controls the various global propagation properties. The listed values apply to all security servers unless you override the defaults by specifying different values in the [secsrv_name] section for a given security server.
- The [secsrv_name] section lists each security server in accordance with your propagation hierarchy. The listed values apply only to the specified server, where secsrv_name is the fully qualified domain name (FQDN) of the security server. You must configure the [secsrv_name] section for each security server in your realm in order to identify its parent-child relationships in the propagation hierarchy and override one or more default values for a given server. This is an optional parameter.

Following is a brief description of the kpropd.ini file sections:

The [default_values] Section

You cannot override the interval, service_name, or primary_realm values that you set in the [default_values] section. In other words, the values you set for these parameters in the [default_values] section override any other value you assign to them in the subsequent [secsrv_name] sections.

Following are the options in the [default_values] section:

`interval = n [s|m|h|d]`

Specifies how often to propagate database changes to the other security servers, where n indicates the number of seconds, minutes, hours, or days. The default value is 15 seconds.

NOTE

Intervals less than 15 seconds may generate a lot of network traffic during peak authentication.

`key_exp=n[s|m|h|d]`

Specifies the length of time for which a session key is valid, where `n` indicates the number of seconds, minutes, hours, or days.

The default is value 6 hours.

`max_cache=n[K|M]`

Specifies the maximum size that each cache file of the security server (`prop_hostname`) can reach before it is deleted, where `n` indicates the number of bytes, kilobytes, or megabytes. A deleted cache file initiates a full database propagation when the connection is re-established.

The default value is 1024 kilobytes.

`max_retry_delay=n[s|m|h|d]`

When `kpropd` attempts to establish a connection with a secondary security server and the attempt fails, `kpropd` waits for a period of time called the retry delay, initially set for 1 minute. With each subsequent timeout, the retry delay doubles. The `max_retry_delay` is the maximum interval between retries that `kpropd` must wait before it terminates its attempt to establish a connection with a secondary security server and logs the failure to the system log.

`net_timeout=n[s|m|h|d]`

Specifies the length of time the propagation system waits for a response from any security server before terminating the connection, where `n` indicates the number of seconds, minutes, hours, or days. When a timeout occurs, all propagating records are cached into the `prop_hostname` file associated with the target server. When a connection to the server is re-established, records in the cache file are then propagated.

The default value is 30 seconds.

`port=port_name` Specifies the communication port over which the database is propagated. The value can be a well-known service or a numeric value, but must be listed in the `/etc/services` file. The default port is `kerberos-adm`.

`primary_realm=DEFAULT_REALM`

Specifies the default realm of the primary security server.

If the `krb.conf` file does not exist, the `DEFAULT REALM` is assigned the uppercase equivalent of the domain name.

`realms=[all|realm1[, realm2][,...]]`

Specifies the realms whose records are propagated to the secondary security servers. The default value, `all`, propagates principal records from all realms to all security servers.

`service_name=service_principal_name`

Specifies the name of the service principal with access to the propagation system on the local security server, where `service_principal_name` is the name of the service principal. The default value is `host/fqdn@REALM`, where `fqdn` is the FQDN of the host, and `REALM` is the realm name of the host.

The [secsrv_name] Section

The `secsrv_name` section is the FQDN of the security server specific to this section.

Following are the options in the `[secsrv_name]` section:

`parent=fqdn` Specifies the parent security server of the `secsvr_name` in the propagation hierarchy, where `fqdn` is the FQDN of the parent server. Every security server except the primary security server must have only one parent server. The primary security server has no parent server.

NOTE You cannot override the `interval`, `service_name`, or `primary_realm` values that you set in the `[default_values]` section.

`child[n]=fqdn` Specifies the child security server of the `secsrv_name` in the propagation hierarchy, where `fqdn` is the FQDN of the child server. A security server can have zero or more child servers.

If more than one child server receives propagated records from `secsrv_name`, include a complete child configuration line for each additional child, where each child is uniquely numbered with the suffix `n`, beginning with `child1`.

NOTE

You cannot override the `interval`, `service_name`, or `primary_realm` values that you set in the `[default_values]` section.

Examples

Following is a sample `[secsrv_name]` section in the Kerberos configuration file:

```
REALM1
REALM1 secsrv1.company.com admin server
REALM2 secsrv1.company.com admin server
REALM1 secsrv1.company.com
REALM2 secsrv2.company.com
REALM2 secsrv2.company.com
REALM1 secsrv3.company.com
REALM2 secsrv4.company.com
```

The `[secsrv_name]` section denotes a propagation hierarchy where `secsrv1` is the primary security server and the parent of the secondary security server, `secsrv2`. In addition, `secsrv2` is the parent of the `secsrv3` and `secsrv4` secondary security servers.

The security servers `secsrv1` and `secsrv2` support the realms `REALM1` and `REALM2`. The secondary security server `secsrv3` supports only `REALM1`, while `secsrv4` supports only `REALM2`. All servers have a `host/fqdn` principal in `REALM1`.

Following is a sample `[default_values]` section:

```
[default_values]
interval=15s
key_exp=6h
max_cache=1024K
max_retry_delay=1h
net_timeout=30s
port=kerberos-adm
primary_realm=REALM1
realms=all
service_name=host
[sersrv1]
child = secsrv2
[secsrv2]
child1 = secsrv3
child = secsrv4
parent = secsrv1
[secsrv3]
parent = secsrv2, realms = REALM1
[secsrv4]
parent = secsrv2, realms = REALM2
```

The `[default_values]` section lists the default values that `mkpropcf` may create using the `krb.conf` file on a primary security server, which supports `REALM1` as its default realm. The propagation hierarchy that `kpropd` creates is derived from the security servers that support the default realm.

Because the `krb.conf` file cannot describe a propagation hierarchy, where secondary security servers themselves have secondary security servers, edit the `kpropd.ini` file to support such relationships.

The prpadmin Administrative Application

The `/opt/krb5/adm/prpadmin` administrative application runs on all security servers and helps you manage the propagation system. For example, to propagate all the contents of the primary principal database to all the secondary security servers, execute the `prpadmin full_dump` command on the primary security server.

You can run `prpadmin` from the command line or in a shell-like command loop.

If you run `prpadmin` from the command line, you must specify a particular command, as shown in the following example:

```
#./prpadmin full_dump
```

To run `prpadmin` in a shell-like command loop, execute `prpadmin` without any command, and specify a particular command, as shown in the following example:

```
#./prpadmin > full_dump
```

Typically, commands have the format `command [hostnames]`. On the primary security server, the only valid host names belong to secondary security servers; on the secondary security servers, the only valid host name is that of the primary security server or those secondary security servers that act as child servers to the parent secondary security server. Specify host names as FQDN.

For more information on `prpadmin`, type `man 1 prpadmin (1)` at the HP-UX prompt.

Setting Up Propagation

After installing and configuring your primary and secondary security servers, you must propagate principal database information from the primary security server to all secondary security servers.

Before you can configure propagation, each secondary security server must have an existing principal database to act as a container for the information being propagated to the server. The principal database is created during installation.

Each security server must also have a stashed master key. If you created the database during installation, the key is automatically stashed in the `/opt/krb5/.k5.REALM` file. If you created the database after installation using `kdb_create`, verify that you stashed the key using the `kdb_create -s` or `kdb_stash` tool.

The `mkpropcf` tool aids propagation configuration by reading the Kerberos configuration file, `krb.conf`, and constructing the required propagation settings.

The primary security server component contains three daemons. You need to restart and stop these daemons at various times throughout propagation.

NOTE During initial propagation of the principal database to all secondary security servers, the startup order of the services or daemons is critical. However, after the initial propagation is complete, the startup order is irrelevant.

Table 9-2 lists the daemons, and briefly describes their functions. To avoid confusion and redundancy in this section regarding names, Table 9-2 also indicates the generic names used in this document to discuss the daemon.

Table 9-2 primary security server Services and Daemons

Daemon Name	Function	Generic Usage Name
`kadmind`	Accepts administration requests from the administrator and database propagation requests from the propagation daemon, `kpropd`.	Admin
`kdcd`	Accepts authentication requests from client applications	KDC
`kpropd`	Initiates the principal database propagation process	Prop

To propagate the principal database entries on the primary security server to secondary security servers, complete the following steps:

NOTE Some steps are specific to the primary security server or to the secondary security server. Ensure that you perform the task on the appropriate server.

Step 1. Create the propagation configuration data using the following steps:

1. Log on to the primary security server as root.
2. On the primary security server, verify that the correct information is added to the `krb.conf` file to locate all secondary security servers and to identify the correct default realm of the primary security server.

3. From the primary security server `/opt/krb5/install` directory, run the following command:

   ```
   # mkpropcf
   ```

   This creates the `kpropd.ini` file, which defines your propagation hierarchy.

NOTE

If you do not want to use the default hierarchy structure (a two-tier system), you must edit the `kpropd.ini` file to implement your preferred hierarchy. For more information on this file, see "The kpropd.ini File" on page 251.

4. Copy the `kpropd.ini` file to the secondary security server.
5. If you have configured a multitiered hierarchy, that is, if you have secondary security servers that act as propagation parent servers, copy the configuration file of the primary security server to each secondary security server.

Step 2. On the primary security server, add the `admin` principal and extract the service key by using the following command:

```
# /opt/krb5/admin/kadminl -R <admin/principal name> <passwd>
```

Step 3. Extract the propagation principal on the primary security server, using the following command:

```
# /opt/krb5/admin/kadminl -R ext <service principal name>
```

By default, `host/fqdn@REALM` is added.

Step 4. Kill all the running daemons on the secondary security server, and extract the service key by using the following commands:

```
# /sbin/init.d/krbsrv stop
```

```
# /opt/krb5/bin/kadmin <admin/principal> -R ext <service princ
ipal name>
```

NOTE The <admin/principal> is the same as the one added on the primary security server in step 2.

Step 5. Start the admin daemon on the secondary security server by using the following command:

```
# /opt/krb5/sbin/kadmind
```

Step 6. Start the propagation daemon on the primary security server, and verify whether the daemons have started or not, by using the following command:

```
# /opt/krb5/sbin/kpropd
```

```
Propagation System Started.
[fqdn] subscriber_ip_addrs: <nnn.nnn.nnn.nnn>
[fqdn] Established secure connection for
propagation
[fqdn] no cache file or OK file found; will do
full dump
[fqdn] starting full dump
[fqdn] full_dump successful.
```

where:

fqdn Specifies the FQDN of the secondary security server.

nnn.nnn.nnn.nnn Specifies the IP address of the secondary security server.

If the primary security server propagates to multiple secondary security servers, the message sequence is displayed for each secondary security server.

Step 7. Start the kdcd and the propagation daemon on the secondary security server, and verify whether the daemons have started, by using the following commands:

```
# /opt/krb5/sbin/kdcd
```

```
# /opt/krb5/sbin/kpropd
```

Verify that propagation has occurred on the secondary security server by using the `kdb_dump` utility to view the contents of the principal database on the secondary security server. The existence of recently added principal accounts indicates a successful propagation.

For information on troubleshooting problems encountered during propagation, see “Monitoring Propagation” on page 263.

Monitoring Propagation

You must regularly monitor database propagation between servers. Monitoring helps you to identify the following problems:

- Primary-secondary link failure
- Stalled propagation

To monitor the propagation, you need to examine the log file and the propagation queue files.

When propagation problems occur, the copies of the database on the secondary security servers do not match with the database on the primary security server. See “Comparing the Database to Its Copies” on page 265 for more information on detecting and resolving the mismatch condition.

For troubleshooting the resolution problems, see Chapter 11, “Troubleshooting,” on page 289.

Monitoring the Log File

You can use the log file to identify a secure connection failure between the primary and secondary security server. The syslog file contains the problems that occur while propagating data or while establishing a secure link between the servers for propagation.

`$KPROPD` is an unique header that identifies the errors generated by the propagation daemon, `propd`, in the syslog file. You can create a cron job to parse the log file at regular intervals and notify the security administrator or the system owner, for example, via paging or e-mail, when a critical error message is found.

Critical Error Messages

The following error messages indicate critical problems:

```
Authentication failed: hostname server error: error_name

Can’t find kpropd.ini registry key/file.

[hostname of peer] Can’t establish secure connection for
propagation (errno=error_name); connect delay is seconds sec
connect delay is seconds sec
```

```
[hostname of peer] Can't connect to subscriber to
propagate principal database information
[hostname of peer] could not get service ticket
[hostname of peer] full_dump failed
[hostname of peer] not enough memory to allocate work buffer
Not enough free system resources to run or start the
propagation system.
Propagation system aborting.
Not enough system resources free to read from propagation
queue
Propagation system aborting.
Out of memory
```

For more information on resolving the problems, see "Propagation Failure" on page 269.

Monitoring Propagation Queue Files

The propagation queue file, `prop_q`, contains the database changes that occur on the primary security server. The `kpropd` daemon reads the `prop_q` file that lists all principals that have changed since the last propagation cycle. At the end of a successful propagation cycle, all the security servers have up-to-date principal databases.

To indicate successful propagation, `kpropd` creates a zero-length file, `prop_`*`hostname`*`_ok`, where *`hostname`* specifies the security server to which the data is propagated. If the propagation fails, a `prop_`*`hostname`* file is created and all the unpropagated changes are saved to the file. After propagating these changes to the secondary security servers, the contents of the queue file, `prop_`*`hostname`*, are deleted. If the `prop_`*`hostname`* or the `prop_`*`hostname`*`_ok` file exists for a specific host, `kpropd` dumps the entire primary database to the secondary security server without a queue file.

Monitoring Old File Date and Large File Size

In rare cases, a propagation failure or stall may occur without indicating an error message in the syslog file. Undertake additional monitoring measures to check for proper functioning of the propagation system. You can monitor the propagation queue files for unusual characteristics, such as old file creation date or large file size. Under normal conditions, these files are created, deleted, and appended many times in a day. For example, if a `prop_q.wrk` file exists with a file creation date older than 24 hours from current time, or if the `prop_q` file size is unusually large, the propagation cycle may be stalled.

For example, a `prop_`*`hostname`* file that is older than 48 hours or is unusually large indicates a propagation problem between the primary and secondary security servers as specified in *`hostname`*.

Updating the principal.ok Time Stamp

You may notice that, by default, the time stamp of the `principal.ok` file on the primary security server does not update after propagation. On the primary security server, the `principal.ok` file retains the date when the database is initially created. Do not change this database unless the database is deleted and rebuilt.

On the secondary security server, the `principal.ok` file is updated each time a propagation full dump from the primary is successfully completed. During a full dump, the database updates are copied to temporary files on the secondary security server. When all of the temporary files are updated, they are put into place on the secondary security server as permanent database files. If a full dump is not completed, the principal database files on the secondary security server, including the `principal.ok` file, remain unchanged, and the date does not change.

Because most propagation occurs as incremental propagation, an old time stamp on the `principal.ok` file of the secondary security server does not necessarily indicate a propagation failure, whereas a new `principal.ok` file indicates a failure.

Comparing the Database to Its Copies

No built-in functionality is available in the principal database to automatically ensure consistency between the database on the primary security server and the secondary security servers. However, you can use `/opt/krb5/admin/kdb_dump` to manually compare the contents of two databases and to identify discrepancies.

You must periodically check that the primary and secondary databases are synchronized. An out-of-sync condition is likely to exist when the following events occur:

- **Authentication problems**

  The out-of-sync condition may first appear as an intermittent authentication failure. In this scenario, a principal that changes the password after the password expires is unable to authenticate, even though the password change is successful. The principal continues its attempt to authenticate, and succeeds if the authentication

attempt is sent to the primary security server. However, if the principal fails on one server as many times as specified by the `MaxFailAuthCnt` parameter in the password policy file, that principal is locked out.

NOTE

HP authentication servers do not issue different messages for different situations that cause authentication failure. For security reasons, the error message displayed is the same for bad password, bad user, or locked user.

Situations such as incorrectly typed password or locked users, which cause authenticaton failure, are not sufficient to indicate an out-of-sync condition.

- **Administration appears normal**

  An out-of-sync condition can also occur when the administration appears to be normal. When a principal that has changed the password fails to authenticate, the principal reports the problem to the administrator. The administrator uses one of the administration tools to unlock the user and change the password of the user to a simple value. Further, the administrator provides the new password to the user.

  The principal may fail to authenticate with the new password and reports the problem to the administrator. The problem is not solved in spite of reporting the problem repeatedly, which can indicate that the databases are out-of-sync and the propagation has stopped. A principal unable to authenticate multiple times can also indicate an out-of-sync problem.

- **Log files indicate problems**

  An indication of the propagation failure in the primary and secondary security server log files provides a clue to the out-of-sync problem. If `kpropd` is not running on the primary security server and each secondary security server, you can be certain that an out-of-sync condition exists.

- **Mismatch between the number of principals**

  Both the machines must contain the same number of principals. Few discrepancies can exist if the database is dumped during a propagation cycle; only a few principals may differ due to an

incremental database propagation. To ensure accurate results, dump the databases simultaneously when administrative activity is at a minimum. Under these conditions, consider a discrepancy of more than five principal entries to be significant.

- **Authentication test to the primary security server succeeds, but fails on the secondary security server**

  The last step to confirm the out-of-sync problem is to force authentication tests to go to the primary security server. You only need to do this for one or two machines. Ensure that the test principal is not locked and that you know the password. Edit the `krb.conf` file and comment the secondary security server entries by placing a hash symbol (#) in the first column on each secondary security server entry.

  Following is a `krb.conf` file in which the secondary security server entries are commented out:

```
#FINANCE.BAMBI.COM fnc01.bambi.com
#IT.BAMBI.COM it02.bambi.com
NETWORK.BAMBI.COM netwrk05.bambi.com admin server
```

  You need to authenticate from the machine with the new configuration file. If authentication succeeds continuously, you have your final clue that the out-of-sync condition exists.

The kdb_dump Utility

To view details of any discrepancy between a primary and secondary principal database and to look for out-of-sync conditions, export each database to a text file and compare the text files. You can dump the databases by stopping the daemons or services and then using the `kdb_dump` utility. You must stop the daemons before using `kdb_dump`.

To identify the difference between the primary and secondary security server database, complete the following steps:

Step 1. On a secondary security server, stop the daemons and execute the following command at the HP-UX prompt:

```
# /opt/krb5/admin/kdb_dump -f /tmp/secondary.db
```

Step 2. From the primary security server, stop the daemons and execute the following command at the HP-UX prompt:

```
# /opt/krb5/admin/kdb_dump -f /tmp/primary.db
```

Step 3. Restart the daemons on both the primary and secondary security servers.

Step 4. To compare the files for discrepancies, copy the files to a common location and execute the following command at the HP-UX prompt:

```
#

diff primary.db secondary.db > diffs_p.db
```

The `diff` command creates a file, `diffs_p.db`, which lists each principal entry on the primary database that does not match an entry on the secondary principal database. For more information on the `diff` command, type `man 1 diff` at the HP-UX command prompt.

Restarting Propagation Using a Simple Process

To restart the `kpropd`, you can follow either a simple process of restarting the daemons, or a more complex process in which you also create a new propagation queue file and perform a full dump.

To stop and restart the propagation daemons, complete the following steps:

Step 1. On the secondary security server, kill `kpropd` and execute the following command at the HP-UX prompt:

```
# /opt/krb5/sbin/kpropd
```

Step 2. On the primary security server, kill `kpropd` and execute the following command at the HP-UX prompt:

```
# /opt/krb5/sbin/kpropd
```

Restarting Propagation Using the Full Dump Method

You can also restart propagation by clearing the propagation directory and restarting `kpropd`, which starts a full dump of the database to all secondary security servers.

To initiate a full database dump to all the secondary security servers, complete the following steps on the primary security server:

Step 1. Stop the propagation daemon by running the `kill` command.

Step 2. Remove the propagation queue files by running the following command:

```
# rm -r -f /opt/krb5/prop/*
```

Step 3. Restart the propagation daemon by using the following command:

```
# /opt/krb5/sbin/kpropd
```

Step 4. Perform a full dump to all secondary security servers by using the following command:

```
# /opt/krb5/admin/prpadmin full_dump
```

This process may take a lot of time if the database contains more than 10,000 principals, and if many secondary security servers exist that act as propagation servers. HP recommends you to initiate this process when the administrative activity is low.

Propagation Failure

If propagation errors occur, complete the following troubleshooting steps:

Step 1. Check that `kpropd` is running on both the secondary and primary security servers. See the “Monitoring Propagation” on page 263 for more information on restarting propagation.

Step 2. Verify that the secret keys for each propagating server are properly extracted to the service key table file. Use `ktutil` to purge any older keys for the `host/principal` from the key table file. If necessary, modify the `host/principal` to re-extract keys, purge older keys from `v5srvtab`, and restart the daemons.

Step 3. Review the `kpropd.ini` file for accuracy. The `kpropd.ini` file must contain the parent-child relationship entries for each security server. If necessary, modify `kpropd.ini`.

Step 4. Verify that the same date and time is set in all security servers. Synchronize time on all the servers to match the primary security server time.

Step 5. Check resource utilization on the server. A 100 percent utilization on a file system prevents `kpropd` from building queue files, which causes propagation to stall or fail. Remove unnecessary files, and archive the log files.

Step 6. Restart the daemons as described in the“Setting Up Propagation” on page 258.

If you encounter the following error message after installing a new secondary security server and attempting propagation, restart the daemons on the secondary security server after the full dump is complete:

```
TGS: Error processing request from host
```

Converting a secondary security server to a primary security server

You may need to convert a secondary security server to a primary security server, for instance, during disaster recovery. During such circumstances, HP recommends that you to reinstall the Kerberos server.

To convert a secondary security server to a primary security server, complete the following steps:

Step 1. Verify that the secondary security server has an up-to-date copy of the principal database. You may need to initiate a full dump of the database from the current primary security server. If your primary security server fails and you cannot perform a full database dump or view the primary log files, review the secondary security server propagation log files to determine the secondary security server that contains the latest copy of the database. Then, copy the `principal.*` files from the secondary security server that contains the recently received propagation data to the secondary security server being converted to the primary. Any changes that are made to the primary database before the failure, but after the last successful propagation, are lost and must be recreated.

Step 2. Retrieve the following files, from the primary security server or from the most recent primary security server backup:

- `/opt/krb5/.k5.REALM`, where REALM is the default realm of the server
- `krb.conf`
- `krb.realms`
- `admin_acl_file`
- `password.policy`
- `kpropd.ini`

Step 3. Archive the `principal.*` files on the secondary security server.

Step 4. Remove the Kerberos server software on the secondary security server.

Step 5. Install the Kerberos server software on the previous secondary security server. Do not create the database during installation.

Step 6. Restore the `principal.*` database files archived in step 3.

Step 7. Restore the original files retrieved from the primary security server in step 2. These are the same files that were created during installation in step 5. Therefore, you will be overwriting them when you restore the original files.

Step 8. Reboot the system, and restart the `kdcd`, `kadmind`, and `kpropd` daemons.

Restarting Services

HP recommends that you restart the services occasionally. It is sufficient to restart the services once a month, unless you have noted problems and want to restart more frequent.

To start a service, run the following commands as a root user:

```
# /sbin/init.d/krbsrv start
# /opt/krb5/sbin/kpropd
```

Cleaning the Temp Directory

Remove files beginning with `krb5_` that are older than a week. If you allow the `tmp` partition to reach 100 percent utilization, the system is unable to write cache files, and the services begin to fail. In addition, check the `/var/tmp directory` for adequate space.

If the number of files beginning with `rc_` are increasing, stop the servers, either by executing the `/sbin/init.d/krbsrv stop` command as a root user or removing files beginning with `rc_`. Restart the servers using the following commands:

```
/sbin/init.d/krbsrv start
/opt/krb5/sbin/kpropd
```

You can execute all these commands with shell scripts or cron jobs.

Configuring Multirealm Enterprises

When you support multiple realms, additional configuration steps are required for both the security servers and clients. This section discusses the servers requirements.

Number of Realms per Database

A single primary security server supports more than one realm. If you have a centralized administration group that controls the security needs of your enterprise, you can support all the realms on one primary security server.

Alternatively, if you have distributed administration groups, you may need to support a single realm for a single primary security server. This arrangement has different configuration requirements.

If you are supporting only one realm per primary security server, you must configure the server normally, and create the required trust relationships, as described in “Configuring Direct Trust Relationships” on page 279.

You must perform additional configuration tasks if you are supporting more than one realm per primary security server.

primary security servers Supporting Multiple Realms

If you choose to support more than one realm in a database of a primary security server, you must decide if all the secondary security servers also support multiple realms. Alternatively, you can have different branches of secondary security servers: one branch for each realm supported in the principal database.

You can configure propagation to propagate only selected realms to a secondary security server. With this propagation configuration, you can maximize the benefits of creating multiple security boundaries in your enterprise. In the event that an authentication server in one branch is compromised, database information about other branches is still secure.

Multiple primary security servers Supporting a Single Realm

You must have one primary security server for each realm if you have distributed administrative groups in which each group maintains its own realm information.

You cannot propagate changes from one primary security server to another. You can only propagate changes from a primary security server to a secondary security server. Therefore, when you have multiple primary security servers supporting only a single database, you do need not to change your propagation configuration from a single-realm scheme.

Adding More Realms to a Multirealm Database

Before adding realms to a database, complete the following steps:

- Install the primary security server and create the database.
- Install each secondary security server and create their respective databases.
- Create the first administrative principal, and assign permissions for all realms of this principal.

In the next section, HP assumes that you have not yet configured propagation before you start adding realms.

To add realms to the database, you can authenticate from a client using the administrative principal account and run the remote administrator, `kadmin_ui`, or you can log on to the primary security server and run the local administrator, `kadminl_ui`.

When you are running administrator, add additional realms using the Realms tab. For more information on creating realms, see “Realms Tab” on page 193.

After adding all the realms to the database, you must decide on the secondary security servers that support multiple realms.

Database Propagation for Multirealm Databases

If you plan to support more than one realm in a single principal database on a primary security server and to propagate only selected realms to certain secondary security servers, you must perform additional steps when you configure propagation.

HP assumes that you are familiar with the propagation setup procedure as specified in “Propagation Hierarchy” on page 243.

You can follow the standard propagation configuration if you have configured a multirealm environment that has only one realm for every primary security server. In other words, if you have multiple primary security servers or if you want to propagate all realms from the primary security server to each secondary security server, complete the following steps:

Step 1. Edit the Kerberos configuration file, `krb.conf`, on the primary security server to contain one entry for each secondary security server that supports a given realm. If a secondary security server supports more than one realm, you must add multiple entries to the file for that server, one for each supported realm. Ensure that you also add one primary security server entry for each realm that the primary security server supports. After you add all the entries, save and close the file.

Step 2. Run the `mkpropcf` utility to create an initial version of the `kpropd.ini` file or registry key.

Step 3. You must edit the `file/registry` key to contain the correct information for your propagation design. For instance, if you want to propagate only certain realms to a selected secondary security server, you must edit the `entry/key` for the parent of that server to indicate only the required realms. For more information on indicating only select realms to propagate, type `man 4 kpropd.ini` at the HP-UX prompt.

Step 4. After configuring the `kpropd.ini` file of the primary security server, follow the propagation configuration steps.

On each Kerberos security server, you need to extract only the `host/key` for the default realm of the primary security server, and not for each realm supported by the secondary security server. Even if the secondary security server does not support the default realm of the primary security server, you must still create a `host/principal` for the secondary security server and extract the key to the key table file of the secondary security server.

10 Managing Multiple Realms

This chapter describes how to set up and configure interrealm authentication between Kerberos servers, and how to manage multiple realms. You must establish trust between the two realms before a principal in one realm can access a service in another realm.

This chapter discusses the following topics:

- “Considering a Trust Relationship” on page 277
- “Configuring Direct Trust Relationships” on page 279
- “Hierarchical Interrealm Trust” on page 281

Considering a Trust Relationship

You can establish a multiple realm environment within your enterprise. Regardless of the reason, if principals in one realm need access to secured services supported in a different realm, you must establish a trust relationship between the realms.

When two distinct realms share secret keys, the two realms are said to trust each another. With that trust in place, principals can securely access services in their native realm as well as those in the trusted foreign realm.

Interrealm authentication begins with relying on a secure authentication between users and the security server in a single realm. The shared interrealm key between trusted servers provides the extra link to create a chain of trust that allows a principal in one realm to authenticate to a service in a trusted foreign realm. To establish a trust relationship, administrators for both realms must have a agreement.

You can configure your Kerberos servers for interrealm authentication based on one-way trust, two-way trust, or hierarchical trust.

One-Way Trust

In interrealm authentication, one-way trust authenticates principals in a realm (Q) to the services in another realm (S), but prevents principals in the realm S from accessing services in the realm Q.

In simple terms, if Harry trusts Sally with his secrets, but Sally does not trust Harry with her secrets, Harry and Sally have a one-way trust relationship between them.

Two-Way Trust

In interrealm authentication, two-way trust authenticates principals in a realm (Q) to the services in another realm (S), and principals in the realm S to the accessing services in the realm Q.

In simpler terms, if Harry trusts Sally with his secrets, and Sally trusts Harry with her secrets, Harry and Sally have a two-way trust relationship between them.

Hierarchical Trust

In interrealm authentication, hierarchical trust allows principals in one realm to access resources in another realm if there is a chain of trust established between the realms. The chain relies on a hierarchical realm naming scheme.

For example, `IT.BAMBI.COM` and `DEER.JUNGLE.COM` are child realms of their respective parent realms, `BAMBI.COM` and `JUNGLE.COM`. If both child realms have two-way trust with the parent realm, and the two parent realms have a direct trust link, `IT.BAMBI.COM` and `DEER.JUNGLE.COM` can have hierarchical interrealm trust between them.

To support hierarchical trust in Kerberos servers, you must have a realm hierarchy, where each realm has a direct relationship with a parent and potentially several children.

Other Types of Trust

You may choose to interoperate with other Kerberos implementations. HP Kerberos server, Microsoft Windows 2000, and MIT Kerberos servers provide Kerberos security solutions following the same IETF standard. HP Kerberos server can interoperate with these other solutions, which allows you to selectively deploy the platforms you choose to meet the needs of your company.

Fore more information on interoperability with Windows 2000, see Chapter 4, “Interoperability with Windows 2000,” on page 51.

Configuring Direct Trust Relationships

If the Kerberos security servers manage all the realms in a multirealm environment, you must add interrealm principals to the principal databases for each realm.

Interrealm principals are special-case `krbtgt/REALM1@REALM2` principal accounts, where `krbtgt/REALM1` is the ticket-granting service principal for realm 1 and `REALM2` is the foreign realm.

A direct trust relationship exists when the server that hosts Realm 1 directly trusts the server that hosts Realm 2.

The client system constructs the interrealm ticket request rather than the servers. Interrealm authentication begins when a user requests a service ticket for a service that is not in the default realm of the user.

The client software constructs the service ticket request, and sends it to the Kerberos server that supports the default realm of the user. Because the service is not in that realm, the Kerberos server cannot return a service ticket. However, if it has a direct trust link to the realm of the service, it can return an interrealm ticket for the realm of the service.

When the client receives the interrealm ticket, it sends the interrealm ticket with the service ticket request to the Kerberos server that supports the realm of the service.

When a foreign Kerberos server receives an interrealm ticket with a service ticket request, and if the interrealm ticket was obtained from a realm where a direct trust relationship exists, the foreign Kerberos server returns the service ticket. For this process to work on the server, the following conditions must be met:

- The user principal must be able to authenticate in the default realm of the user.
- You must establish a trust relationship between the default realm of the user and the realm of the service.

The Kerberos server returns a failure for any of the following reasons:

- The client authentication fails.

- The Kerberos server does not recognize the realm listed in the interrealm ticket, that is, when a proper trust relationship between the realms is not established.
- The Kerberos server does not recognize the requested service principal, and has no further trust relationships for which it returns an interrealm ticket.

To set up a cross-realm authentication between the two realms `ADMIN.BAMBI.COM` and `IT.BAMBI.COM`, you need to create two special principals on each Key Distribution Center (KDC), as shown in the following example:

```
krbtgt/ADMIN.BAMBI.COM@IT.BAMBI.COM
krbtgt/IT.BAMBI.COM@ADMIN.BAMBI.COM
```

This special principal indicates a two-way trust relationship. If you want to configure only a one-way trust relationship, you need to create the following special principal:

```
krbtgt/ADMIN.BAMBI.COM@IT.BAMBI.COM
```

The passwords of the corresponding principals must be the same on both the KDCs. However, the different cross-realm principals do not have to have matching passwords.

For example, `krbtgt/ADMIN.BAMBI.COM@IT.BAMBI.COM` must have the same password on each KDC, but `krbtgt/IT.BAMBI.COM@ADMIN.BAMBI.COM` and `krbtgt/ADMIN.BAMBI.COM@IT.BAMBI.COM` do not have to share the same password.

Hierarchical Interrealm Trust

You need to use hierarchical interrealm authentication when a realm does not have a direct path to its destination realm, but has a path to an intermediate realm.

Hierarchical Chain of Trust

Interrealm trust can be transitive, for example, if realm A trusts B and B trusts C, then a client in A can get a ticket from C by following the trust path from A to B to C.

For example, consider realm 1 as X.Y.A , realm 2 as X.Y.C, and realm 3 as X.Y.B with the following direct trust relationships established between them.

- Realm X.Y.A has a direct trust link to realm X.Y.B.
- Realm X.Y.B has a direct trust link to realm X.Y.C.

In such a configuration, the client walks the realm tree from node X.Y.A to X.Y.C by requesting an interrealm TGT from each intermediate realm (in this example, X.Y.B), until it obtains the service ticket from X.Y.C.

Although creating such hierarchical trusts is more efficient than attempting to configure each server with knowledge of all possible interrealm trust relationships, the client must still perform the realm tree computation, map each realm to a security server host name, and request an interrealm TGT from each realm in the path.

In addition, the Kerberos protocol requires the client to know the exact realm of each service it needs to authenticate to. In the previous example, the client in X.Y.A must know that the service it wants to access belongs to realm X.Y.C.

Assume that a client in the realm RED.BLUE.COM needs to authenticate to a service located in the realm GREEN.YELLOW.COM, but realm RED.BLUE.COM does not have a direct trust relationship established with the realm GREEN.YELLOW.COM.

Now, VIBGYOR.INDIGO.COM has a direct trust relationship established with both RED.BLUE.COM and GREEN.YELLOW.COM. Hence, RED.BLUE.COM can obtain an interrealm ticket through the intermediate realm, VIBGYOR.INDIGO.COM. The client in RED.BLUE.COM requests an

interrealm ticket from VIBGYOR.INDIGO.COM, and can use this interrealm ticket to contact GREEN.YELLOW.COM for a ticket to use a service in its realm.

Hierarchical Interrealm Configuration

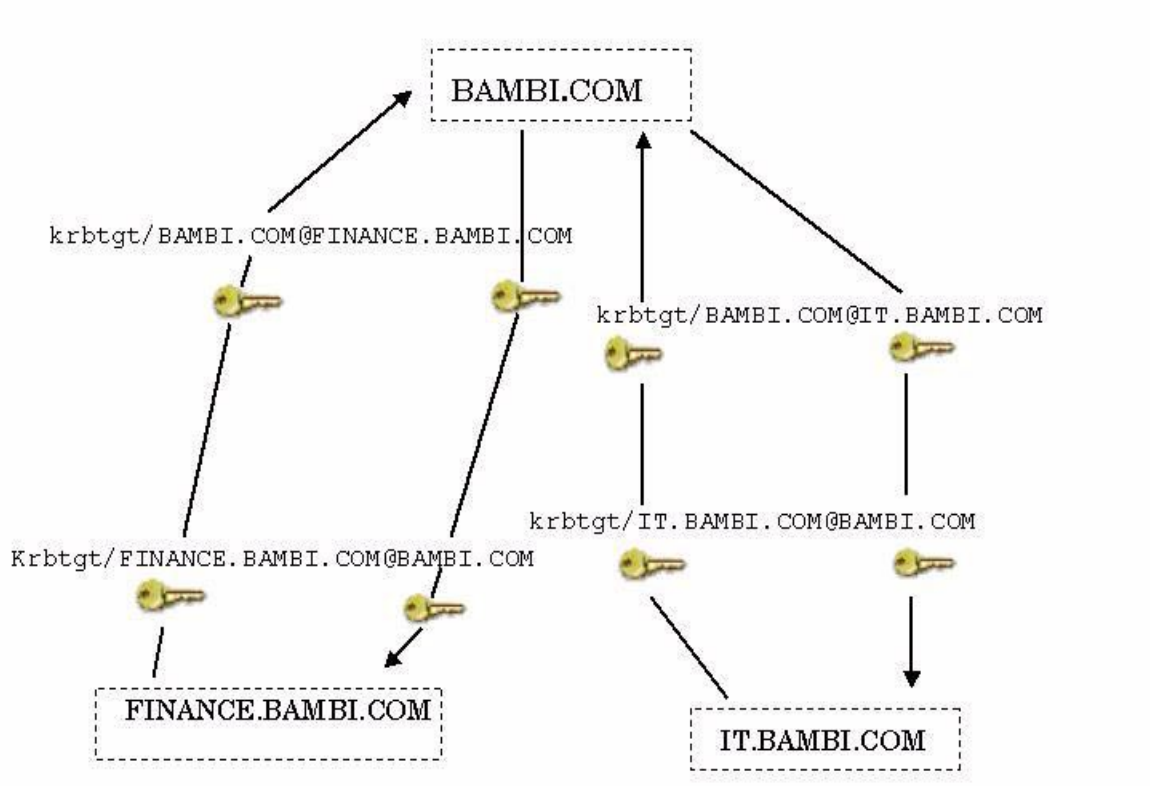
To configure realms to perform hierarchical interrealm authentication, complete the following steps in the local realm, intermediate realm, and target realm:

Step 1. Add an interrealm principal (krbtgt/REALM2@REALM1) to the principal database to allow the local realm to authenticate with the intermediate realm and the intermediate realm to authenticate with another intermediate or the target realm.

Step 2. Add a second interrealm principal (krbtgt/REALM1@REALM2) to the database if you also want the intermediate or target realm to authenticate two-way with the local realm or another intermediate realm.

These actions are described in detail in the following sections. The example configuration in this section uses the interrealm authentication principals shown in Figure 10-1.

Figure 10-1 Hierarchical Interrealm Configuration



The relationships are defined as follows:

- `krbtgt/BAMBI.COM@FINANCE.JUNGLE.COM` allows the server in `BAMBI.COM` to accept tickets from `FINANCE.JUNGLE.COM`.
- `krbtgt/IT.JUNGLE.COM@BAMBI.COM` allows the server in `IT.JUNGLE.COM` to accept tickets from `BAMBI.COM`.

For interrealm authentication in the other direction, two-way hierarchical interrealm authentication, you must also add these principals:

- krbtgt/FINANCE.JUNGLE.COM@BAMBI.COM allows the server in FINANCE.JUNGLE.COM to accept tickets from BAMBI.COM.
- krbtgt/BAMBI.COM@IT.JUNGLE.COM allows the server in BAMBI.COM to accept tickets from IT.JUNGLE.COM.

Configuring the Local Realm

To configure the local realm, consider the local realm as FINANCE.JUNGLE.COM and the intermediate realm as BAMBI.COM and complete the following steps in the FINANCE.JUNGLE.COM realm:

Step 1. Use the Kerberos administrative utility, HP Kerberos Administrator, in the FINANCE.JUNGLE.COM realm, and add the krbtgt/BAMBI.COM@FINANCE.JUNGLE.COM principal, which allows users in the FINANCE.JUNGLE.COM realm to authenticate with the server in the BAMBI.COM realm.

Enable the following settings for this principal:

- Select all the Allow attributes.
- Clear all the Require attributes.
- Provide a password rather than a random key and remember the password.
- Record the primary key type and salt type.
- Record the password key version number.

Step 2. If the FINANCE.JUNGLE.COM realm also trusts the BAMBI.COM realm, add the krbtgt/FINANCE.JUNGLE.COM@BAMBI.COM principal, which allows users in the BAMBI.COM realm to authenticate to the services in the FINANCE.JUNGLE.COM realm.

Step 3. Enable the same settings for this principal as for the interrealm principal, krbtgt/BAMBI.COM@FINANCE.JUNGLE.COM, as mentioned in step 1 in the procedure for configuring the intermediate realm.

Configuring the Intermediate Realm

To configure the intermediate realm, consider the local realm as FINANCE.JUNGLE.COM , the intermediate realm as BAMBI.COM , the target realm as IT.JUNGLE.COM, and complete the following steps in the BAMBI.COM realm:

Step 1. Use the Kerberos administrative utility, HP Kerberos Administrator, to add the krbtgt/BAMBI.COM@FINANCE.JUNGLE.COM principal, which allows users in the FINANCE.JUNGLE.COM realm to authenticate with the server in the BAMBI.COM realm.

Enable the same settings for the principal krbtgt/BAMBI.COM@FINANCE.JUNGLE.COM as used for the principal krbtgt/BAMBI.COM@FINANCE.JUNGLE.COM in the local realm.

NOTE Each intermediate realm has four keys if you are performing two-way interrealm authentication.

Step 2. If the FINANCE.JUNGLE.COM realm also trusts the BAMBI.COM realm, add the krbtgt/FINANCE.JUNGLE.COM@BAMBI.COM principal, which allows users in the BAMBI.COM realm to authenticate with the server in the FINANCE.JUNGLE.COM realm.

Step 3. Enable the same settings for this principal as for the first krbtgt/FINANCE.JUNGLE.COM@BAMBI.COM principal, with the same settings enabled as used for the principal in the local realm.

Step 4. Add the krbtgt/IT.JUNGLE.COM@BAMBI.COM principal, which allows users in the BAMBI.COM realm to authenticate with the server in the IT.JUNGLE.COM realm.

Step 5. Enable the same settings for this principal as for the first krbtgt/IT.JUNGLE.COM@BAMBI.COM principal, with the same settings enabled as used for the principal in the local realm.

Step 6. If the BAMBI.COM realm also trusts the IT.JUNGLE.COM realm, add the krbtgt/BAMBI.COM@IT.JUNGLE.COM principal, which allows users in the IT.JUNGLE.COM realm to authenticate with the server in the BAMBI.COM realm.

Step 7. Enable the same settings for this principal as for the first `krbtgt/BAMBI.COM@IT.JUNGLE.COM` principal, with the same settings enabled as used for the principal in the local realm. Refer to step 2 in “Configuring the Target Realm” on page 286.

Configuring the Target Realm

To configure the target realm, consider the intermediate realm as `BAMBI.COM` , the target realm as `IT.JUNGLE.COM` and complete the following steps in the `IT.JUNGLE.COM` realm:

Step 1. Use the Kerberos administrative utility, HP Kerberos Administrator, to add the `krbtgt/IT.JUNGLE.COM@BAMBI.COM` principal, which allows users in the `BAMBI.COM` realm to authenticate with the server in the `IT.JUNGLE.COM` realm.

Enable the following settings for this principal:

- Provide the same password that you used for `krbtgt/IT.JUNGLE.COM@BAMBI.COM` while configuring the intermediate realm.
- Select all `Allow` attributes.
- Clear all `Require` attributes.
- Record the primary key type and salt type.
- Record the password key version number.

Step 2. If the `BAMBI.COM` realm also trusts the `IT.JUNGLE.COM` realm, add the `krbtgt/BAMBI.COM@IT.JUNGLE.COM` principal, which allows users in the `IT.JUNGLE.COM` realm to authenticate with the server in the `BAMBI.COM` realm.

11 Troubleshooting

This chapter describes how to troubleshoot the Kerberos server, and also includes the strategies and tools to use while investigating the software and hardware components of the Kerberos server.

When you encounter a problem, you may need to investigate many hardware and software components. You can identify and resolve some problems quickly, such as invalid software installation, version incompatibilities, insufficient HP-UX resources, corrupt configuration shell scripts, and programming or command errors. Other problems require more investigation.

After you identify the problem, you can resolve the problem using the suggestions in this chapter or the error messages documented in the link installation manuals. However, there may be problems that you should report to your HP support contact. This chapter also includes guidelines for submitting an HP Service Request (SR).

This chapter discusses the following topics:

- “Characterizing a Problem” on page 291
- “Diagnostic Tools Summary” on page 293
- “Troubleshooting Kerberos” on page 294
- “General Errors” on page 303
- “User Error Messages” on page 305
- “Administrative Error Messages” on page 306
- “Reporting Problems to Your HP Support Contact” on page 308

Characterizing a Problem

You need to consider many questions while trying to characterize a problem. Start with global questions and gradually get more specific. Depending on the response, ask another series of questions until you have enough information to understand exactly what has happened. The key questions to ask are as follows:

- Does the problem seem isolated to one user or program? Can the problem be reproduced? Did the problem occur under any of the following circumstances:
  - — While running a program
  - — While issuing a command
  - — While transmitting data
- Does the problem affect all users? The entire realm? Has anything changed recently? The possibilities are as follows:
  - — New software and hardware installation.
  - — Same hardware but changes to the software. Has the configuration file been modified? Has the HP-UX configuration been changed?
  - — Same software but changes to the hardware. Do you the suspect hardware or software?

It is often difficult to determine whether the problem is hardware-related or software-related. Symptoms that indicate a hardware problem are as follows:

- Intermittent errors.
- Networkwide problems after no change in software.
- Link-level errors, from logging subsystem, logged to the console.
- Data corruption—link-level trace that shows that data is sent without error but is corrupt or lost at the receiver.

Symptoms that indicate a software problem are as follows:

- Network services errors returned to users or programs.

- Data corruption.
- Logging messages at the syslog.

Knowing what has recently changed on your network can also help you understand whether the problem is software-related or hardware-related.

Diagnostic Tools Summary

Table 11-1 describes the most frequently used diagnostic tools, which are documented in the link installation manuals.

Table 11-1 Diagnostic Tools

Tool Name	Description
`netstat`	A nodal management command that returns statistical information regarding your network.
`landiag`	A diagnostic program that tests LAN connections between HP computers.
`ping`	A diagnostic program that verifies the physical connection to a remote host and reports the round-trip communication time between the local and remote hosts. For more information, type `man 1M ping` at the HP-UX prompt.
`nettl`	A utility that traces link-level traffic to and from a node. HP recommends that you enable tracing only when troubleshooting a problem unsolved by other means. The `nettl` utility sends informational messages regarding network activity to the system console or to a file.

Troubleshooting Kerberos

When troubleshooting problems with Kerberos, you need a reference point from which to work. For example, is the problem on the remote system or on the local system? However, the terms “local” and “remote” are limited in their description of complex communications, such as when a local system logs on to a remote system and then the remote system logs back onto the local system. At that point, which is the local system and which is the remote system?

A better solution is to use the terms “client” and “server”. The term “client” refers to a process that requests a service from another process. The term “server” refers to a process or a host that performs operations requested by local or remote hosts that are running client processes.

A typical network service consists of two co-operating programs. The client program runs on the requesting system. The server program runs on the system with which you want your system to communicate. The client program initiates requests to communicate. The server program accepts requests for communication. For example, the network service `rlogin` is a client program that requests a log on to a remote HP-UX or UNIX system. When `inetd` receives the request to log on to the remote host, `inetd` invokes the server program for `rlogin` (called `rlogind`) to handle the service request.

Error Messages

The client or server can generate the error messages generated by a service as seen on the client. Error messages from the client occur before a connection is completely established. Error messages from the server occur after a connection is completely established.

Logging Capabilities

The security server handles the system logging differently.

UNIX Syslog File

The security server daemons, `kadmind`, `kpropd`, and `kdcd`, write error messages to the system log (`/var/adm/syslog/syslog.log`) file. You can also configure the daemons to log the messages in a different file. Use the following command while starting the daemon, to specify a different file name:

```
# kdcd -l <log file name>
```

or

```
# kadmind -l <log file name>
```

However, principal database operations performed locally on the primary security server using the HP Kerberos Administrator are not recorded because these programs do not use syslog to audit their activities.

The syslog daemon (`syslogd`) is configured using the `/etc/syslog.conf` file, which controls where your log files are located. For example, you can configure syslog to send messages to `/usr/adm/messages`.

The security server daemons log an entry for each transaction and whether the transaction succeeded or failed. The number of transactions that are logged in your syslog file is determined by how you have configured the reporting levels.

The security server uses the following syslog reporting levels:

- `LOG_ERR` – Displays security server errors.
- `LOG_WARNING` – Displays security server warnings.
- `LOG_NOTICE` – Displays secured application server errors.

The server logs information messages through `syslog`. The syslog file can grow large if not maintained properly. The `syslog` file is specified in `/etc/syslog.conf`, which has a symbolic link to the `/var/adm/messages` directory.

Check the size of this file to make sure it does not use an overwhelming amount of system disk space. If the `/var` partition grows to 100 percent utilization, syslog stops writing log messages and may even shut down active processes, such as the daemons.

Create a shell script to be executed daily or weekly by cron to check the syslog file size, partition utilization, or both, and to detect any problems. In addition, you must archive the syslog files regularly to a separate partition, drive, or server.

Services Checklist

While troubleshooting ensure, that you have answered all the questions in the troubleshooting checklist in the section “Characterizing a Problem” on page 291. Ensure that your node name and the Internet address exists in the `/etc/hosts` file, and run the service on your own node. If the server is successful in authenticating, the client and the server side of the service operates correctly. This provides a starting point to determine where the problems are occurring.

Troubleshooting Techniques

Table 11-2 describes various troubleshooting scenarios and provides tips for solving potential problems. These tips help you troubleshoot and assist you in determining a problem quickly.

Table 11-2 Troubleshooting Scenarios

Scenario	Cause	Troubleshooting Tips
Permission denied while initializing `krb5`.	The `/opt/krb5/krb.conf` file is set to read-only by root.	Reset the permissions to 644.
Host name cannot be made canonical while selecting the best principal.	The hostname is longer than 8 characters. A `uname -n` returns the first 8 characters of the name.	Add the 8 character name to the `/etc/hosts` file (just tack it on to the end of the current IP address or host name entry).
ASN.1 failed call to system time library while dispatching.	This message is usually displayed when a client is requesting a `krbtgt` with a bad lifetime value.	

Table 11-2 Troubleshooting Scenarios (Continued)

Clock skew too great in KDC reply while getting initial credentials.	This problem generally occurs because the clock of the system deviates too much from the time on the authenticating KDC. A clock skew time of up to 5 minutes is allowed.	You must run NTP or a similar service to keep your system clock synchronize with the atomic clock of the world. If you do not know how to do this, contact your system administrator to resolve this problem.
Requesting host principal without fully qualified domain name. Server not found in the Kerberos database while getting the credentials from KDC. Incorrect network address while getting credentials from KDC.	The host uses the `/etc/hosts` file to resolve name lookups before using DNS. This problem occurs when the entry for the host in the `/etc/hosts` file contains unqualified domain name before the fully qualified domain name. This problem can also occur if the `/etc/hosts` file has a different IP address for a host from what the DNS server has.	
The `/opt/krb5/krb.conf` file not found.	The `krb.conf` file has not been created.	Copy the sample file, `krb.conf.sample`, from `/opt/krb5/example` and edit accordingly.
Cannot open or find the configuration file while initializing the Kerberos code.	This problem occurs when you try to create the database and the `krb.conf` file is not found in the `/opt/krb5` directory.	Copy the sample file, `krb.conf.sample`, from `/opt/krb5/example` and edit accordingly.

Table 11-2 Troubleshooting Scenarios (Continued)

Required parameters in `krb.realms` missing while initializing the Kerberos context.	This problem occurs when the parameters are missing or incorrect in the `krb.realms` file.	Ensure that the `krb.realms` file has the appropriate information.
Stored master key is corrupted while initializing `kadminl` interface.	This message appears if the stash file is corrupted.	Recreate the stash file.
Cannot find or read stored master key while getting the master key.	This problem occurs when the stash file is missing.	Create the stash file
Cannot open or find the Kerberos configuration file while initializing the `krb5` library.	This problem occurs when the `krb.conf` file does not exist.	
Client/server realm mismatch in initial ticket request while initializing the `kadmin` interface.	This problem occurs when you have an old credential cache file, which has credentials for another realm.	Use the `kdestroy` utility to destroy your old credential cache or use the `kadmin -p <pname>` name.
Cannot resolve network address for KDC in the requested realm while getting initial credentials.	Check your `resolv.conf` file.	
Decrypt integrity check failed while verifying the master key.	This problem occurs when the passwords do not match.	
Decrypt integrity check failed while initializing the `kadminl` interface.	This problem occurs when the passwords do not match.	

Table 11-2 Troubleshooting Scenarios (Continued)

Cannot find/read stored master key while getting master key.	This problem occurs when the stash file is not found.	Provide the master key as a command-line option. You can also create the stash file.
Error verifying pre-authentication data type 2.	This problem occurs due to an incorrect password.	Enter the correct password.
Service key not available while getting initial credentials.	This problem can occur if your principal has a 3DES key but not a DES key and the Kerberos client does not support 3DES, or vice versa.	Create identical key types.

Table 11-3 Troubleshooting Scenarios for your LDAP-based Kerberos server

Scenario	Cause	Troubleshooting Tips
LDAP entry type not supported	The Kerberos server does not support reference entries.	Remove the reference entries associated with the Kerberos attribute.
Incomplete Kerberos entry in LDAP	One or more Kerberos principal attributes are missing or corrupted in the Directory server.	Delete the Kerberos principal and recreate it. This could indicate a potential security problem.
Another Kerberos principal associated with this DN	The DN that you have specified for the Kerberos principal is already associated with another Kerberos principal.	Specify another DN as you can associate only one Kerberos principal with any DN.

Table 11-3 Troubleshooting Scenarios for your LDAP-based Kerberos server (Continued)

Scenario	Cause	Troubleshooting Tips
Connection to the LDAP server was lost.	Connection to the LDAP server was lost.	Verify that the Directory server is accessible, else restart the Directory server. You can also restart the Kerberos server, if needed.
LDAP server timeout	The directory server timed out a request.	You may need to increase the time out on the Directory server.
LDAP server error when processing the request	The Directory server encountered an error while processing the request.	Look into the corresponding error on the Directory server for troubleshooting this. You may need to change the parameters like maximum number of entries returned on the Directory server.
Improper format of `/opt/krb5/krb5_ldap.conf`	The format of the configuration file `/opt/krb5/krb5_ldap.conf` is incorrect	Verify the contents of the entries present in the configuration file `/opt/krb5/krb5_ldap.conf`

Table 11-3 Troubleshooting Scenarios for your LDAP-based Kerberos server (Continued)

Scenario	Cause	Troubleshooting Tips
LDAP authentication failed	The Kerberos server was unable to connect to the Directory server with the information provided in the `/opt/krb5/krb5_ldap.conf` configuration file.	Verify that the values of the `proxy_user` and `proxy_user_password` are correct. Ensure that you change the value of `proxy_user_password` when you change the either the password of the proxy user in the Directory server or change the master key.
LDAP initialization failed		Verify that the Directory server's hostname and port number are valid
`ldap_clientauth_init` failed		Verify that the certificate present in the `certdbpath` is valid. Verify that the Directory server has to be properly configured to use SSL.
`ldap_init` failed		Verify that the certificate present in the `certdbpath` is valid. Verify that the Directory server has to be properly configured to use SSL.

Table 11-3 Troubleshooting Scenarios for your LDAP-based Kerberos server (Continued)

Scenario	Cause	Troubleshooting Tips
LDAP database is read-only	An attempt to modify the Kerberos entry failed as the Directory server entry is read-only.	Edit the Kerberos configuration file, `krb5_ldap.conf`, to specify a directory server that can be updated and restart all Kerberos server applications
Insufficient access on LDAP	The proxy user does not have sufficient privileges to add, modify, delete, and search for entries on the Directory server.	Change the configuration on the Directory server to allow add, modify, delete, and search privileges under the `default_princ_subtree` and `base_dn_for_search`. When you add a Kerberos principal ensure that you specify it under the `base_dn_for_search`.
Incorrect LDAP DN	The DN specified is not valid.	Ensure that you add the DN under the `base_dn_for_search` in the Directory server.
Unavailable or invalid `libldap.so`		Verify that the LDAP-UX product is installed correctly on the Kerberos server.

General Errors

Following are the general errors that you may encounter while setting up your Kerberos server:

- Ensure that the Domain Name Server (DNS) is working properly. Several aspects of Kerberos rely on this name service. It is important that your DNS entries and your hosts have the correct information. The canonical name of each host must be a fully qualified host name, including the domain, and each IP address of the host must resolve to the respective canonical name.
- Ensure that you remove all trailing spaces in the configuration files. Trailing spaces can cause problems with the server. If trailing spaces are present in the configuration file, the following error message appears:

  ```
  kdcd cannot start the database for the realm
  ```

- The Kerberos daemons `kdcd` and `kadmind`, by default, do not dump core.

  If you, as the administrator, want the `kadmind` daemon to dump core, you need to create a `DEBUG` file in the directory `/var/adm/krb5/kadmind/DEBUG`, with `setuid bit` set.

  If you need the `kdcd` daemon to dump core, you must create a `DEBUG` file in the directory `/var/adm/krb5/kdc/DEBUG` with the `setuid bit` set.

Forgotten Passwords

If a user forgets the password, you must reset the password. To reset the password, you must have the following correct administrative permissions:

- `i` for Inquire About Principals.
- `c` for Change Principal Passwords.

Using the graphical user interface or the command-line administrator, change the password and inform the user of the new temporary password. By default, the user must change the password on the next logon.

Locking and Unlocking Accounts

If a user or a service principal exceeds the maximum number of failed authentication attempts allowed by the password policy file, the account is locked and the principal is not issued a ticket. Alternatively, a security administrator may have purposefully locked a principal account so that it cannot be used. In each case, the principal remains in the principal database but is unable to use the Kerberos services.

To unlock a principal account, use the graphical user interface or command-line administrator. In the HP Kerberos **Administrator>Principal Information>Principals** tab, clear the **Lock Principal** checkbox.

You must have the correct administrative permissions (`i` for Inquire About Principals and `m` for Modify Principals) to lock or unlock an account.

Invoke the command-line administrator, `kadmin` and use the `mod [principal] attr {lock | unlock}` command.

Clock Synchronization

While client clocks are not required to be closely synchronized with the security server or application server, HP recommends that you loosely synchronize all client clocks with the server.

If the client clock is outside the permitted clock skew of 5 minutes, the log file on the client system will contain the entries that indicate the condition.

To eliminate the warnings, synchronize the client clock with the server to within 5 minutes.

NOTE You must closely synchronize all security server and application server clocks. HP recommends that you implement a secured time service to ensure that all clocks are synchronized.

User Error Messages

Users may see error messages while using the Kerberos server. The following sections describe user error messages, explain their causes, and suggest corrective actions.

Decrypt Integrity Check Failed

Explanation: This message is displayed if a user requests a ticket from the server and one of the following conditions is true:

- User entered the wrong password for the realm chosen for the ticket request. A different password for each realm may exist that the user is permitted to access.
- User entered an incorrect principal name during logon.
- User principal account has been locked out of the security network and is not authorized to receive tickets from the server.

This message may also appear when a user attempts to change his or her password. In this case, these conditions apply to the entries in the `kpasswd` file of UNIX clients.

Depending on how you have configured the `MaxFailAuthCnt` parameter in the password policy file, the user may have had a sufficient number of failed authentication attempts to be locked from obtaining another ticket.

Action: Unlock the principal account using an administrative tool.

Password Has Already Been Used or Is Too Close to Current One

Explanation: This error message appears if a user chooses a password that he or she has previously used. The maximum number of previous passwords to compare the new proposed password against is defined in the `MaximumHistory` entry in the password policy file. The default is `1`.

Action: Instruct the user to choose a password that has not been used previously.

Administrative Error Messages

Following are some messages that administrative principals may see when using their accounts. This section also contains some recommended solutions.

Password Has Expired While Getting Initial Ticket

Explanation: This message may appear when a user tries to log on as a remote administrator using the remote command-line administrator, `kadmin`. This implies that when the principal account was created, it was configured to require a password change. This is the default behavior. Alternatively, it may occur if the administrative principal password has expired.

Remote command-line administrator does not permit password changes. This error does not occur when using the local administrator because the local administrator does not require a password to be entered.

Action:

If you set the `Change Password Required` attribute, use the local administrator, `kadminl_ui`, to disable the `Require Password Change` option for the administrative principal to be used for remote administration.

- If you are using the HP Kerberos Administrator, select the **Principal Information>Attributes** tab and clear the **Require Password Change** checkbox.
- If you are using the command-line administrator, `kadmin`, use the `mod [principal]` command and set `nopwchg` to indicate that password change is not required.

Service Key Not Available While Getting Initial Ticket

Explanation: This message may appear when a user tries to log on to the remote administrator. It also may occur while using the `kinit` and `kpasswd` utilities. It means that the user is trying to use a key type that is different than the one assigned to the user in the principal database, in other words, DES versus 3DES. By default, the HP Kerberos Administrator and command-line administrator are set to use a 3DES

key during authentication. If the principal does not have a 3DES key, the tools attempt to negotiate a supported key type. If the tools cannot negotiate a supported key type, the error message `Service key not available while getting initial ticket` is returned.

Action

If the user is using the command-line administrator, verify that the user does not enter an inappropriate key type during logon with the `-e` switch. Because both the HP Kerberos Administrator and command-line administrator negotiate the best key type, it is not necessary to use the `-e` switch to specify a key.

To change the key type assigned to the principal, edit the principal settings in the HP Kerberos Administrator>Password tab.You cannot change the key type of the principal using the command-line administrator.

Reporting Problems to Your HP Support Contact

If you do not have a service contract with HP, you may follow the procedure described below but you will be billed accordingly for time and materials.

If you have a service contract with HP, document the problem as a Service Request (SR) and forward it to your HP support contact. Include the following information where applicable:

- A characterization of the problem.

  Describe the events leading up to and including the problem. Attempt to describe the source of the problem. Describe the symptoms of the problem.

  Your characterization of the problem must include the following: HP-UX commands, communication subsystem commands, job streams, result codes and messages, and data that can reproduce the problem.

  Illustrate as clearly as possible the context of any message. Prepare copies of information displayed at the system console and user terminal.

- Obtain the version, update, and fix information for all software.

  To check your Kerberos server version, execute the `what` *`binary_name`* command, where the *`binary_name`* indicates a Kerberos server binary, such as `kadmin, kdcd`, or `kadmind`.

  To check the version of your kernel, execute `uname -r`.

  This allows your support contact to determine if the problem is already known, and if the correct software is installed at your site.

- Record all error messages and numbers that appear at the user terminal and the system console.

- Save all network log files.

  Prepare the formatted output and a copy of the log file for your HP support contact to further analyze.

- Prepare a listing of the HP-UX I/O configuration you are using for your HP support contact to further analyze.
- Try to determine the general area within the software where you think the problem exists. Refer to the appropriate reference manual and follow the guidelines on gathering information for problems.
- Document your interim or “workaround” solution. The cause of the problem can sometimes be found by comparing the circumstances in which it occurs with the circumstances in which it does not occur.
- Create copies of any Internet Services or other trace files that were active when the problem occurred for your HP support contact to further analyze.
- In the event of a system failure, take a full memory dump. Use the HP-UX utility `/sbin/savecore` to save a core dump. See the *HP-UX System Administration Tasks* manual for details. Send the output to your HP support contact.

A Configuration Worksheet

The following worksheet helps you configure your Kerberos server with LDAP as the backend database.

Table A-1 Configuration Worksheet

Configuration Worksheet for LDAP database	
Directory administrator DN	
Directory server host	
Directory server port	
Base DN for search	
Subtree DN	
Proxy user DN	
Certificate db path **NOTE:** Enter the Certificate db path only if you plan to use SSL as your security mechanism	

Following is an explanation and sample table.

Table A-2 Configuration Worksheet Explanation

LDAP Server Configuration Worksheet	
Directory administrator DN	The distinguished name of a directory administrator allowed to modify the schema on the Directory server. Example: `cn=directory manager`
Directory server host	The host name or IP address where your directory server is located. Example: `deer.bambi.com (18.43.56.117)`
Directory server port	The TCP port number your directory server is using. Example: `389`

Table A-2 Configuration Worksheet Explanation (Continued)

LDAP Server Configuration Worksheet	
Base DN for search	The default base DN for search is the root of the directory tree on the Directory server, where the Kerberos server searches for kerberos principals. Example: `ou=people, o=bambi.com`
Default Principal Subtree DN	The default principal subtree DN is where all Kerberos principals are added by default, if no LDAP entry is specified while creating the kerberos principal. The default principal subtree DN must be located under the default base DN for search. Example: `ou=accounts, ou=people,o=bambi.com`
Proxy user DN	The distinguished name of the proxy user, if needed. Example: `cn=proxyuser, ou=speical users o=bambi.com`
Certificate db path	The location of the database that contains the certificates for your client. Example: `/.netscape`

B Sample krb.conf File

The sample `krb.conf` file named `krb.conf.sample` is available in the `/opt/krb5/examples` directory. Copy this sample file to `/opt/krb5/krb.conf` file and modify it to reflect the host names and realm name for your realm.

NOTE If you have configured your Kerberos server with C-Tree as the backend then the realm names are case sensitive. If you have configured your Kerberos server with LDAP as the backend then the realm names are not case sensitive.

Replace the underlined `Your_Realm_Name`, `Your_Secondary_Server1`, `Your_Secondary_Server2`, and `your_primary_server` with the name of your Kerberos `REALM`, secondary security servers, and the primary security server host names, respectively.

Use the following format to write a sample `krb.conf` file:

```
Your_Realm_Name
Your_Realm_Name Your_Secondary_Server1
Your_Realm_Name Your_Secondary_Server2
Your_Realm_Name your_primary_server admin server
```

Following is an example of the krb.conf file:

```
BAMBI.COM
BAMBI.COM fox.bambi.com
BAMBI.COM goat.bambi.com
BAMBI.COM deer.bambi.com admin server
```

where:

- The realm name is `BAMBI.COM`.
- The secondary security server 1 is `fox.bambi.com`.
- The secondary security server 2 is `goat.bambi.com`.
- The primary security server is `deer.bambi.com`.

For more information, type `man 4 krb.conf` at the HP-UX prompt.

The services File

The `services` file contains entries that allow client applications to establish socket connections to the KDC or to the applications servers. A KDC client requires the following entries in the `/etc/services` file:

```
#
# Kerberos services
#
kerberos5     88/udp kdc         # Kerberos V5  kdc
kerberos      88/tcp kdc         # Kerberos V5  kdc
kerberos-adm 749/tcp             # Kerberos 5 admin/changepw
krb5_prop     754/tcp            # Kerberos slave propagation
kerberos-cpw  751/tcp kerberos_master # Kerberos changepw
```

NOTE If you do not use the `krbsetup` tool to configure your Kerberos server, you must manually add the services mentioned in “The services File” on page 317 to the `/etc/services` file.

For more information, type `man 4 services` at the HP-UX prompt.

C Sample krb.realms File

The sample `krb.realms` file named `krb.realms.sample` is available in the `/opt/krb5/examples` directory. You can copy this sample file to the `/opt/krb5` directory, and modify it to reflect your realm name.

NOTE The realm names are case sensitive.

Replace the underlined `Your_Realm_Name`, `Your_Primary_Security_Server`, `Your_Secondary_Server_Server`, and `Your_Domain_Name` with the name of your Kerberos `REALM` and host names of the primary security server and secondary security servers.

```
Your_Primary_Security_Server Your_Realm_Name
.Your_Secondary_Security_Server Your_Realm_Name
*.Your_Domain_Name Your_Realm_Name
```

The following is an example of the `krb.realms` file with a brief explanation:

```
admin.bambi.com BAMBI.COM     # map host directly
.fox.bambi.com BAMBI.COM      # all hosts in domain
*.bambi.com BAMBI.COM         # all the other hosts belonging
                                to the domain and subdomains
```

- The first line of the `krb.realms` file maps the host `admin.bambi.com` to the `BAMBI.COM` realm.
- The second line of the `krb.realms` file maps all hosts in the `fox.bambi.com` domain to the `BAMBI.COM` realm.

NOTE The preceding dot in this line identifies the first field as a domain name rather than a host name.

Typically, this line is not required as the realm name. By default, is the upper-case equivalent of the domain name.

- The third line of the `krb.realms` file maps all hosts in the domain and subdomains with the root name `bambi.com` to the `BAMBI.COM` realm.

Glossary

A-B

admin_acl_file (administrator access control list) Text file that lists the administrators and their respective permissions.

HP Kerberos Administrator The graphical user interface that is used to administer the principal database of the Kerberos server.

Authentication Service (AS) Authentication is a verification of a user's identity. The Authentication Service (AS) hands out a ticket-granting ticket, which in turn is used to access the Ticket-Granting Service (TGS).

C

Command-line administrator Command-line utility used to administer the Kerberos principal database.

F

ftp Client application that securely copies files to and from remote computers via the File Transfer Protocol.

ftpd Secured daemon that services connection requests from secured and nonsecured `ftp` clients.

K

.k5.REALM A hidden file that contains your master key, where `REALM` is your default realm.

k5login A hidden ASCII file that contains a list of principals authorized to access a specific user account.

kadmind Daemon that accepts administration requests from remote administrative tools.

kdb_create Creates a principal database and adds a realm to an existing principal database.

kdb_destroy Securely removes the principal database.

kdb_dump Copies the contents of the principal database to `stdout` or to a text file.

kdb_load Loads a database with the principal entries from a database dump text file.

kdb_stash Stores the master key, the transformed master password, to a disk file.

KDC The heart of the Kerberos realm is the Kerberos Key Distribution Center (KDC), which is also generally referred to as the Kerberos Server. The KDC is composed of two different services the Authentication Service (AS) and the Ticket-Granting Service (TGS). The KDC issues Kerberos tickets.

kdcd Daemon that provides two services: the Authentication Service (AS) and the Ticket-Granting Service (TGS).

Key Distribution Center See KDC.

kpropd daemon that propagates the principal database from one server to another.

kpropd.dbg File that stores the output from the principal database propagation system when the `prpadmin` program runs in debug mode.

kpropd.ini Propagation configuration file `mkpropcf` creates using information in the local `krb.conf` file.

krb.conf File that contains configuration information that describes the default realm of the host, the administration server, and security servers for known realms.

krb.realms The realms file defines host-to-realm or domain-to-realm name mapping data. The `krb.realms` file is located only on the Kerberos server systems. This file maps host names to realm names.

M

mkpropcf Creates the `kpropd.ini` file, which is the default propagation configuration file in a propagation hierarchy.

P

password.policy Text file that controls the password policy for all principals.

principal Files that make up the principal database, including `principal.dat`, `principal.idx`, and `principal.ok`.

prop_hostname ASCII cache file created on the propagating security server when the host name security server is unavailable during the principal database propagation cycle. The file contains principal records that could not be propagated from the `prop_q` file to the receiving server.

prop_hostname_ok Empty file created on the propagating security server after a successful propagation to the host name security server.

prop_q Default propagation input queue file that contains the names of every principal whose record has changed since the last successful database propagation.

prop_q.wrk Temporary working copy of `prop_q`, the default propagation input queue file.

prpadmin Application used to administer database propagation between servers.

R

rlogin Application that provides log on access to remote systems.

realm Each administrative domain have its own Kerberos database, which contains information about the users and services for that particular site or administrative domain. This administrative domain is the Kerberos realm.

T

telnet Application that connects to a remote host using the `telnet` protocol.

telnetd Daemon that accepts connections from secured and nonsecured `telnet` clients.

ticket A certificate issued by an authentication server, which helps verify the user's identity to any given service.

TGS The Ticket-Granting Service (TGS) authorizes the user to gain access to the network resources by generating and issuing tickets.

TGT Issues tickets to connect to hosts in its own domain.

Ticket-granting ticket See TGT.

V

v5srvtab Binary file that contains service principal names and their corresponding secret keys.

Symbols

#, 68
/etc/rc.config.d/krbsrv, 102
/opt/krb5/sbin, 69
/sbin/init.d/krbsrv start, 102
/var/adm/krb5/krb5kdc, 315

A

access control list
  See ACL
ACL, 112
adding a realm, 273
ADMD = 1, 102
admin_acl_file, 64, 101
  # comment, 113
  format, 113
  identifier, 113
  instance, 113
  perms_list, 113
  using wildcards, 113
administrative server, 112
AS, 321
ASN.1 failed, 296
authentication, 27
auto-configure, 69
automated tool, 69

B

bad key
  example, 228

C

case sensitive, 316
Cerberus, 25
child server, 243
client, 294
clock skew, 297
clock synchronization, 304
configure the slave KDC, 103, 105
core dump, 309

D

3DES encryption type, 31
data corruption, 291
database propagation, 274
DEBUG file, 303
default key name, 227
default propagation interval, 243
default@REALM principal, 125
default_values section, 252
DES encryption type, 31
DNS, 303
Domain Name Server
  See DNS
domain-to-realm, 66, 93
domain-to-realm mapping, 322
dump file, 43

E

/etc/hosts file, 296
/etc/inetd.conf file, 37
/etc/syslog.conf file, 295
Edit the Configuration Files, 92
encryption type, 44
  3DES, 31
  DES, 31
error message, 294
extracted key, 179

F

file/registry key, 274
FQDN, 66, 127, 297
fully qualified domain name
  See FQDN

G

good key, 228

H

hierarchal trust, 278
host service principal, 104
host/fqdn@REALM principal, 126
hostname.subdomain.domain.com, 316, 320
host-to-realm, 66, 93
host-to-realm mapping, 322
HP Kerberos Administrator, 111
HP-UX 11i v2 operating system, 37

I

IETF, 25, 278
incorrect net address, 297
incremental propagation, 248
inetd daemon, 294

initial ticket, 26
intermediate realm, 285
intermittent error, 291
Internet Engineering Task Force
  See IETF
interrealm authentication, 275
issuing a ticket, 322

K

/krb5/admin_acl_file, 112
K/M key name, 44
K/M@REALM principal, 124
kadmin/changepw@REALM principal, 126
kadmin/REALM@REALM principal, 126
kadmind daemon, 104, 227
kadminl utility, 227
kcpwd/REALM@REALM principal, 126
kdb_create -s, 96
kdb_create utility, 228
kdb_migrate command
  -M option, 44
kdb5_util command, 43
KDC, 321
KDC = 1, 102
kdc.conf, 66
kdc.conf file, 322
kdc.conf.sample, 319
kdcd daemon, 227, 303
Kerberos
  client, 130
  database, 322
  realm, 322
  reference material, 26
Kerberos daemons, 102
kerberos daemons, 102
Kerberos V5 protocol, 28
kpropd daemon, 246, 274
kpropd.ini, 64
kpropd.ini file, 246, 274
  sections
    default_values, 252
    secsrv_name, 252
krb.conf, 64, 65
krb.conf file, 112
krb.conf Format, 65
krb.realms, 64, 66
krb.realms file, 112
krb.realms Format, 67
krb5.conf file, 65
krb5.conf.sample, 315
krbsetup, 69
krbsetup tool, 45, 48
krbtgt/principal, 31
krbtgt/REALM@REALM principal, 125, 279
ktutil, 99

L

landiag tool, 293
link-level error, 291
local administrator, 111
local client, 130
local realm, 284
logging an error message, 293

M

master key name, 44, 227
MaxFailAuthCnt parameter, 305
Microsoft Windows 2000, 278
MIT Kerberos server, 278
mkpropcf tool, 246

N

netstat tool, 293
nettl tool, 293
network service, 37
non-kerberos service, 37

O

/opt/krb5/kadmind/DEBUG directory, 303
one-way trust, 277

P

parent server, 243
password change request, 112
password policy file, 101
password.policy, 64
password.policy file, 42
ping tool, 293
pound sign (#), 68
prop_hostname file, 246
prop_hostname.ok file, 246
prop_q file, 246
prop_q.wrk file, 246
propagation, 272
prpadmin tool, 247

R

remote administrator, 111
remote request, 112
reporting level, 295
  LOG_ERR, 295
  LOG_NOTICE, 295
  LOG_WARNING, 295
RFC 1510, 22, 25, 54
RFC 1964, 22, 54
RFC 2743, 22
RFC 2744, 22

S

sample kdc.conf, 319
sample krb.conf, 315
Sample krb.realms, 319
sample krb5.conf, 315
secsrv_name, 252
server, 294
service contract, 308
service key table, 130
Service Request
  See SR
service ticket, 27
services file, 317
services manpage, 317
session key, 27
setuid bit, 303
shell% /opt/krb5/sbin/kadmind, 102
shell% /opt/krb5/sbin/krb5kdc, 102
software distributor
  reference material, 39
Software Selection dialog box, 39
Specify Source window, 39
SR, 290, 308
start the kerberos daemons, 102
stash file, 227
support contact, 290
swinstall command, 39
syslog file, 295

T

Table 3-1, 64
target realm, 286
TGS, 321, 322
TGT, 26, 27
ticket flags, 28
ticket-granting ticket
  See TGT
tracing, 293
transitive interrealm trust, 281
troubleshooting
  checklist, 291
  tools, 293
trust, 275
two-way trust, 277

U

uname command, 308
  -r option, 37
unified administration interface, 130
unlocking a principal, 304

V

/var/adm/krb5/kdc/DEBUG directory, 303
/var/adm/messages directory, 295
/var/adm/sw/swagent.log file, 39
/var/adm/syslog/syslog.log file, 295

Y

Your_Realm_Name, 316, 320